Commodore INFO



INCL 22 PAGINA's AMIGA

Prijs ƒ 7,95 / Bfr. 160

8 710966 001332

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS

Jaargang 6, NO.8, dec. '89 / jan. '90

LISTINGS:

Lettertekenaar 64
Aanvallen 64
Bowlen 64
Spreadsheet 64
Music maker 128

THE FINAL CHESSCARD
HANDYSCANNER 64
VOORRAAD 128
DTV OP DE AMIGA
AMIGA C CURSUS
INTEGRATED GEOS
AMIGA HARDDISKS GETEST

Vaste rubrieken:

Tips en Trucs voor de C-64
Amiga Tips
Amiga C
GEOS Machinetaal
Kleine advertenties

Commodore Info

Verschijnt 8x per jaar
Jaarg.6, no.8, dec/jan. 1989

Uitgave: Sala Communications

Uitgever: Vic Sharfman

Redactie:
Ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdred.
drs. J. Boers eindredacteur
H. Smeenk, drs. U. Schuurmans, R. Goudriaan, B. Munniksma, B. Venema, P. Boncz, MGCC/Johan & Johan

Produktie
drs. H. Zoete
M. van Kaathoven J. Broekhuizen

Redactiesecretariaat: R. van Zalingen
Strip: Bert Tier
Illustraties: Ben van Mierlo

Advertentie-exploitatie:
Ing. V. Sala, Ing. B. Sala,
D. van Vlijmen
Postbus 43048
1009 ZA Amsterdam
tel. 020-273198

Redactie adres:
Postbus 43048
1009 ZA Amsterdam
tel. 020-228871

Listingtelefoon: (ma: 17.00-21.00 u)
02155-25162

Abonnementen en administratie:
Sandra v.d. Meyden en Marjo Jansen
Postbus 43048
1009 ZA Amsterdam
tel. 020-248006

Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.

Abonnement:
Voor 8 nummers *f* 47,50 of Bfr. 975 per jaar. Betaling op giro 4985259 (België: BBL nr. 310050602562) t.n.v. SAC/Commodore-Info. Oude nummers kunt U alleen krijgen bij vooruitbetaling van *f* 6,75 op de bovenstaande rekening. Ook telefonische opgave voor een abonnement is mogelijk. Bel GRATIS 06-02242222 (teleservice), elke dag tot 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze telefoonnummers zijn alleen bedoeld voor opgave van NIEUWE abonnementen.
Opzegging dient schriftelijk te geschieden uiterlijk twee maanden voor de aanvang van een nieuwe abonnementsperiode van een jaar.

Zetwerk & druk: NDB, Zoeterwoude
Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

ISSN: 0169-3085

Inhoud van dit nummer

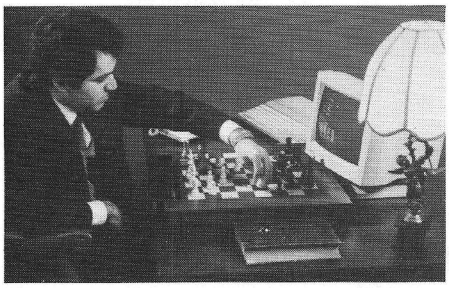
C-64 Handyscanner 6

Het scannen van allerlei illustraties en ook teksten beleeft de laatste jaren een grote opgang. Ook voor de C-64 is scannen mogelijk. We bespreken de Handyscan 64.

Graphics op de 64 8

In onze serie Graphics voor de 64 zijn we alweer aan het zesde deel toe. Dit maal begeven Michel de Boer en Hylke Sprangers op het artistieke vlak: Computerkunst.

The final Chesscard 14



Beschrijving van een zeer intelligente schaakcartridge, die uw computer tot een tegenstander van allure maakt.

Oud van Goudriaan 18

Rob Goudriaan bekeek weer een aantal programma's uit het recente verleden, die nog steeds de moeite waard zijn.

Tips & Trucs 64 20

De vaste rubriek met wetenswaardigheden voor de C-64-gebruiker.

Basic 8 24

Eindelijk is er een basicpakket wat de 80 cls mode ondersteund. Johan & Johan vertellen u het fijne ervan.

Voorraad 128 26

Een bespreking van een C-128 softwarepakket voor voorraad beheer.

Amiga C cursus 28

Na enkele maanden van absentie is de C-Cursus weer terug.

Integrated GEOS 51

Een korte cursus GEOS Integratie leert u de vele mogelijkheden van GEOS programma's toe te passen.

Geos Machinetaal(7) 39

Alweer de zevende aflevering van deze serie, met ditmaal aandacht voor de listing van QuickTop.

Amiga Harddisk 61

Zeker voor het meer professionele werk is een harde schijf voor de Amiga onontbeerlijk.

Inzake invoer 63

Alle ins en outs van de invoer via het toetsenbord met de Amiga.

DTV op de Amiga 70

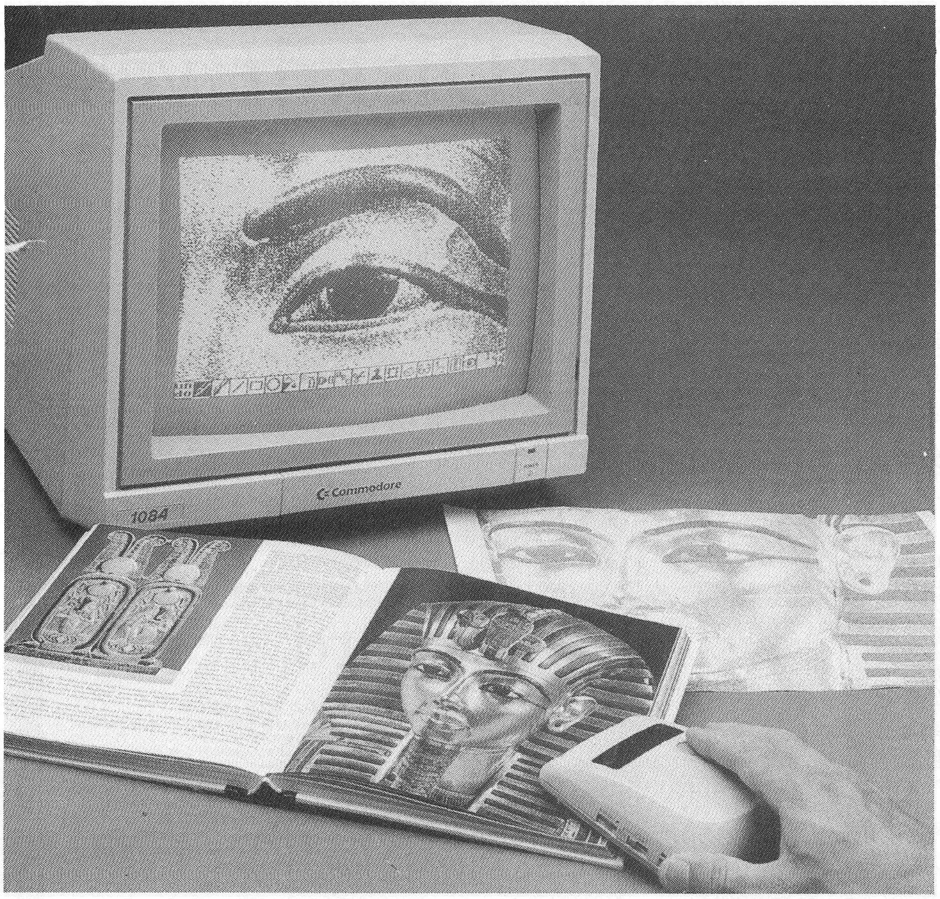


Een uitgebreid artikel over de vele mogelijkheden die de Amiga biedt op het terrein van DeskTop Video. In dit artikel is ook een lijst opgenomen van de beschikbare software en hun dealers.

Redactioneel: Horizontaal en Vertikaal

Commodore is met de Amiga de laatste tijd een richting op gegaan, vooral in Europa, die misschien wat minder aansluit bij wat er traditioneel in Commodore Info met betrekking tot de Commodore computers werd gepubliceerd. Commodore zoekt steeds meer de vertikale markten op, waar bijvoorbeeld met de grote Amiga's aan professionele geluids- en beeld produkties wordt gewerkt. De Amiga is daar natuurlijk bij uitstek voor geschikt, en de steeds snellere en professionelere modellen in de 2000 serie bieden een uitstekend platform. Wij besteden bijvoorbeeld in dit nummer utgebreid aandacht aan de mogelijkheden van de Amiga op het gebied van Desktop Video. Ook op de drukke Amiga '89 show in Keulen was de aandacht voor professionele toepassingen opvallend.
De hobby-gebruiker van de Amiga zal bij deze ontwikkeling niet altijd voordeel hebben, want de prijzen voor allerlei apparatuur kunnen aan de hoge kant zijn als ze specifiek voor de professionele markt zijn bedoeld. Aan de andere kant zullen de mogelijkheden van de Amiga-lijn enorm zijn, en welke actieve hobbyist wil niet graag met professionele apparatuur werken? Commodore heeft beloofd dat de ontwikkelingen op het gebied van het besturingssysteem voor iedereen beschikbaar zullen komen, het is nu alleen te hopen dat de hobby-Amiga niet helemaal wordt vergeten en er ook op het gebied van de 500 gewoon door wordt ontwikkeld.

Jan Boers



Handyscanner64

Al een groot aantal jaren wordt de Commodore 64 op de markt gebracht. Sinds jaar en dag wordt op grote schaal software voor deze computer gekopieerd. Dat dit voor de leveranciers die de legale software op de markt brengen steeds moeilijker wordt om dit op een verantwoorde wijze te doen is begrijpelijk. Eén van de leveranciers die dit goed heeft aangevoeld, en de software onder goede begeleiding en service op de markt brengt is de firma CATRONIX uit Capelle aan de Ijssel. Uit hun grote assortiment voor de Commodore 64 hebben we ditmaal uitgekozen de Handyscanner.

Een handyscanner lijkt op een muis, werkt als een muis, voelt aan als een muis, maar is het niet. De scanner bestaat uit een rij lichtjes en fotocellen welke de teruggekaatste informatie doorgeven aan de computer. Kort weg gezegd, de handyscanner is een optische lezer die het mogelijk maakt plaatjes van een stuk papier in te lezen in een tekenprogramma. Om dit mogelijk te maken wordt de computer uitgebreid met een cartridge. Deze uitbreiding bevat de hardware die nodig is om deze uitbreiding goed te laten werken. Doordat dit buiten de computer om gebeurt blijft een zeer groot gedeelte van het 'dure' geheugen vrij voor programmatuur.

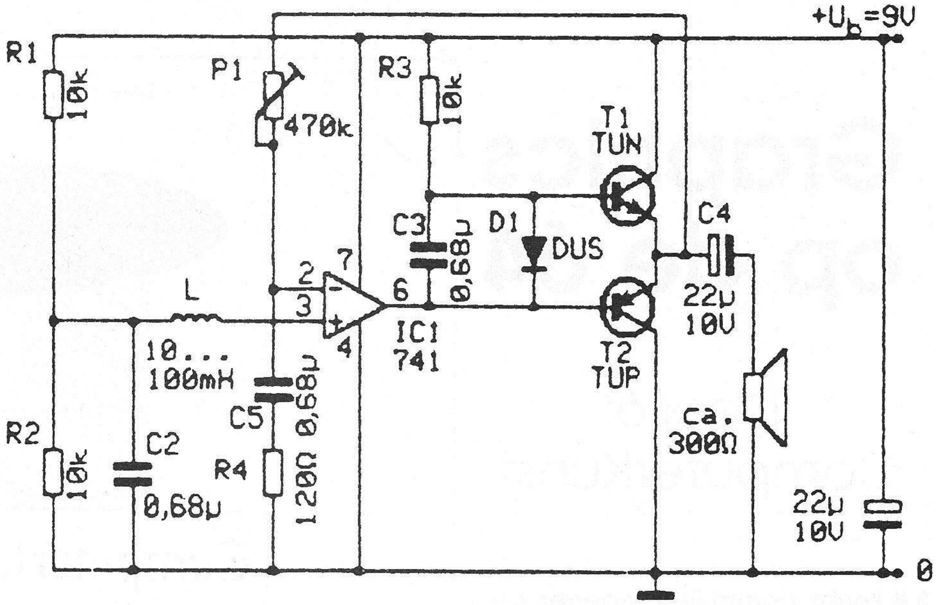
Bij het uitpakken van het pakket is het eerste dat opvalt de prima verpakking. Het wordt geleverd in een grote doos waarbij alle kwetsbare onderdelen goed verpakt zitten tussen lagen schuimrubber. Het gehele pakket bestaat uit een aantal onderdelen. Een handzame scanner die via een kleine plug verbonden wordt met de cartridge. Deze Cartridge wordt gebruikt via de userpoort van de Commodore 64. Door het gebruik van een aantal schuifjes is het niet mogelijk dat de cartridge er verkeerd wordt ingestoken. Let er wel altijd op dat de computer en alle nageladen. Onder aan het tekenscherm komen de menu funkties. Deze zijn te gebruiken met de joystick of met een muis. randapparatuur uit staat. Bij een ingeschakelde computer is de kans groot dat er onderdelen worden beschadigd. De cartridge heeft een externe voeding. Dit is gedaan om de spanning die de Commodore zelf gebruikt niet extra te belasten. Deze spanning wordt verkregen door de meegeleverde trafo. Verder zijn daarbij natuurlijk ook de benodigde software en natuurlijk niet te vergeten de handleiding bijgesloten. De Handyscanner is gemaakt in Duitsland door de firma Scantronik.

Het gebruik

Zijn alle onderdelen goed aangesloten, dan kan de computer aangezet worden. De scanner is nu bedrijfs klaar. Door de knop boven op de scanner kort in te drukken wordt de scanner in de bedrijfs stand gezet. Dit is te zien doordat de lampjes wat feller gaan branden. Door nogmaals de knop in te drukken wordt deze handeling weer te niet gedaan. In deze mode-wordt er dan ook geen data ingelezen. Aan de zijkant van de scanner bevinden zich ook een aantal regelaars. Aan de linkerzijde drie (volgens de handleiding maar twee) en aan de rechterzijde één. Hiermee kunnen worden ingesteld het contrast, zwart/wit of een kleuren origineel, de breedte en het aantal dots per mm. (voor de Commodore 64 alleen mogelijk met de stand 8). Wat bij een bepaald origineel het beste resultaat geeft is alleen door ervaring of met experimenteren goed te bepalen. Het geheel is nu alleen nog niet te gebruiken, eerst moet de nodige software worden geladen. De software bestaat uit een diskette met daarop een aantal programma's. Het eerste programma is een uitgebreid teken programma. Met behulp van dit tekenprogramma kunnen beelden via de scanner worden ingeladen. Na het opstarten niet meteen de diskette uit de diskdrive halen, er wordt nog een gedeelte voor de scanner

Aktiveert U nu de scanroutine, U weet wel door het indrukken van de knop boven op de scanner, dan verschijnt op het scherm de vraag wat de vergrotingsfaktor moet zijn. Hierbij moet U ingedachte houden dat 100% een één op één beeld is, dus het origineel is gelijk aan de kopie. De vergrotingfaktor kan varieren van 30% (3x zo klein) tot 300% (3x zo groot). Deze instelling heeft ook invloed op het aantal beeldpunten dat wordt gebruikt. Bij een vergroting van 300% wordt de maximale breedte gebruikt, dit is 640 beeldpunten. Bij 100% is dit 200 punten. Na het ingeven van de faktor wordt het gehele scherm blauw. Er wordt nu gewacht op data die door de scanner worden doorgegeven. Eerst wordt de scanner op de juiste plaats gezet, dit is goed te zien door het venster. Aktiveren, en langzaam de scanner verschuiven, ongeveer 4 cm

per seconden. Tijdens deze handeling is er op het beeldscherm een flikkering te zien. Hierna moet er even worden gewacht, de data wordt vertaald met een snelheid van 400.000 Bits per seconde. Het geheel duurt bij maximale vergroting ongeveer 20 seconden. Het menu bestaat uit een aantal blokjes met daarin ikonen. Ikonen zijn grafische symbolen die een funktie voorstellen. Gekozen kan er worden door middel van de muis of de joystick. Dit gebeurt door het aanwijzen van het juiste blokje en dan op de knop te drukken. De balk onderin het scherm is het eerste menu, dit is echter nog niet alles. Handyscan heeft nog meer van deze menubalken., in het totaal zelfs drie stuks. Met de muis kiest U deze menubalken door het indrukken van de linker muis knop. Met de joystick gebeurt dit door helemaal onder aan het scherm te gaan staan en dan op de vuur knop te drukken. Voor mensen die veel met een muis werken kan hier iets belangrijks zijn opgevallen, meestal zitten de knop funkties precies anders om. Ook hieraan is gedacht, de gebruiker kan in dit geval aangeven dat de muis knoppen precies andersom moeten werken. De gevorderde gebruiker kan al deze menu funkties veel sneller uitvoeren door het indrukken van bepaalde toetscombinaties. Dit werkt, zeker voor de opties die regelmatig worden gebruikt, vele malen sneller. Het is ondoenlijk om alle funkties van het tekenprogramma hier op te noemen. Het programma doet niet onder voor vele, vaak dure, concurrenten. Scrollen, zoomen, spiegelen, vergroten, koördinaten gebruiken, raster mogelijkheden alles is aanwezig. Het gehele beeldscherm is 640 bij 400 beeldpunten. Dit zijn vier beeldschermen. Hier kan door middel van de ikonen door heen gescrold worden, dit gaat heel langzaam, sneller is het door het indrukken van een van de toetsen 1-4, waarbij 1 linksboven is en 4 recht-onder. Het beeld springt nu naar het juiste scherm. Het geheel kan worden bekeken door het indrukken van de overzichts ikoon. Een groot voordeel vindt ik dat uit het tekenprogramma, zonder dat een reeds getekend beeld verloren gaat, er disk funkties kunnen worden uitgevoerd. Bijvoorbeeld tijdens het werken blijkt dat de schijf vol is, er kan nu tussen door een schijf worden geformatteerd. Bepaalde onderdelen van een tekening zijn er uit te halen en deze als stempel te bewaren. Een goed voorbeeld hiervan is de file, die ook op de meegeleverde disk staat "Analog.bs". Hier zijn onderdelen van een elektrisch schema als los staande onderdelen weergegeven. Door het oppakken , spiegelen, vergroten enz. is het mogelijk geworden complexe elektrische schema's te tekenen. Nu is het zelfs mogelijk hier teksten bij te schrijven. Een hulpmiddel om alles hier goed op zijn



plaats te zetten is het raster. Op deze manier is ook schema 1 getekend. Het raster wordt niet mee afgedrukt, het is slechts een (onmisbaar) hulpmiddel.

Printen

Het print onderdeel is een zeer belangrijk onderdeel bij het tekenprogramma. Dit is meestal de zwakste schakel, is de printer niet goed, of niet goed afgesteld, dan kan ook het uiteindelijke resultaat niet goed zijn. Hiervoor staat er op de programma diskette een printer aanpassings programma. Hier in kunnen alle specifieke gegevens van onze printer worden aangegeven. Dat een 24 naalds printer meestal het beste resultaat geeft mag als bekend worden verondersteld. Dit gaat alleen op als de software deze driver ondersteund, en dat is hier gelukkig wel het geval. Ook de papier lengte wordt hier aangepast. De instellingen worden naar de disk weggeschreven, Alle Commodore printers en Epson compatibel printers worden ondersteund. Pas bij het printen van een tekening worden deze gegevens van de disk geladen. Deze instellingen, zij staan in de file "HSCN+" kunnen ook op de datadisk worden weggeschreven. Dus eerst aanpassen en daarna gaan tekenen. De variabele instellingen, de kantlijnen, de linefeed en de afdruk kwaliteit moeten worden ingegeven na de keuze printen.

Pagefox

Het meest uitgebreide Desk Top Publishing pakket voor de Commodore 64 is ongetwijfeld Pagefox. Ook dit software pakket wordt door CATRONIX geleverd (299,—). Een goede software leveranciers stapt niet van zijn produktlijn af en dus is de Handyscanner compatibel met pagefox. Dit houdt in dat de scanner zonder meer gebruikt kan worden in het DTP pakket. Dit maakt het werken met dit pakket er veel eenvoudiger op, tekeningen kunnen op papier worden gemaakt en probleem loos tussen de tekst worden opgenomen.

Conclusie

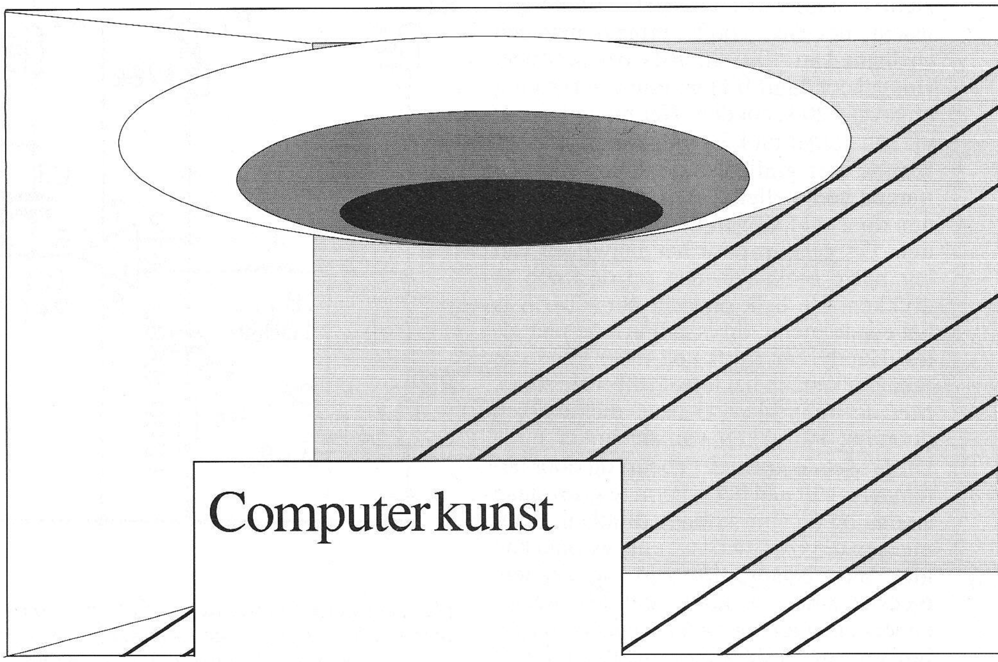
Pc gebruikers zijn er intussen al aan gewend, vele computeraars zouden er niet meer buiten kunnen, tekeningen eenvoudig naar de computer over te brengen. Tot een half jaar terug was dit voor de amateur niet te betalen. De geheugen ruimte van de Commodore 64 was tot nog toe te klein om deze scanners ook hier te gebruiken. Uit Duitsland is dan toch nog een oplossing gekomen in deze Handyscanner. Catronix heeft met het uitbrengen van deze scanner niet alleen een prima stukje hard- software op de Nederlandse markt gebracht, maar ook voor de eerder geleverde software is dit een prachtige aanvulling.

Prijs*f* 599,— inclusief B.T.W.
Leverancier: Catronix Capelle aan de Ijssel
Inlichtingen: 010-4507696

RBG

Graphics op de 64

Deel 6: ComputerKunst



Waren computers vroeger alleen getallenkrakers, die een eindeloze reeks getallen als uitvoer leverden; tegenwoordig zijn computers uitgerust met high-resolution schermen, waarop de prachtigste computer graphics gemaakt kunnen worden. Door deze ontwikkeling is de computer een nieuw medium voor kunstenaars geworden. Met dit nieuwe medium kunnen geheel nieuwe vormen van kunst gemaakt worden, zoals bijvoorbeeld bewegende kunst.

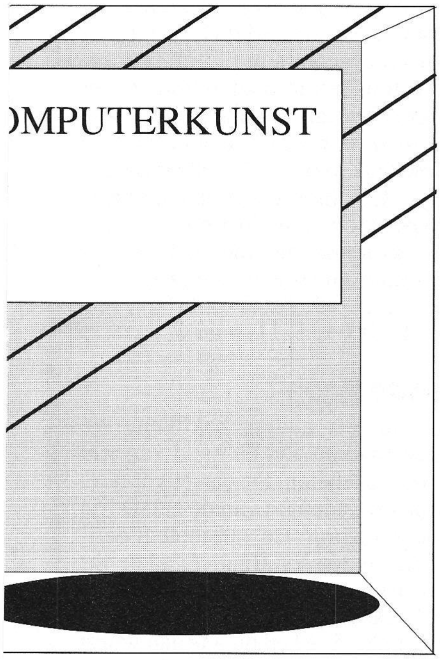
In de vorige afleveringen hebben we een basis gelegd voor het werken met grafische objecten op de Commodore 64. Deze afleveringen waren enigszins theoretisch van aard, met name de afleveringen over high resolution. Ter afwisseling van de theorie zullen we in deze aflevering een begin maken met ComputerKunst. Daarbij spitsen we ons toe op het maken van grafische plaatjes op het scherm. De figuren die op het scherm worden getekend, worden allemaal aan de hand van wiskundige formules berekend. Het is echter niet onze bedoeling om alle wiskundige formules tot op de bodem uit te werken. We willen meer laten zien wat er allemaal mogelijk is met computer graphics. Het uitgebreid uitwerken van de wiskundige formules zou het artikel te lang en onleesbaar maken. We zullen vooral kleine programma's geven, waarmee geëxperimenteerd kan worden.

Omzetting Simon's Basic naar Hires Routines	
Simon's Basic	Hires Routines
HIRES X,Y	POKE 53265,PEEK(53265) OR 32 POKE 53272,PEEK(53272) OR 8 SYS 49264:SYS 49225,Y,X
PLOT X,Y,MODE	SYS 49162,X,Y,MODE
LINE X1,Y1,X2,Y2, MODE	SYS 49186,X1,Y1,X2,Y2,MODE
TEST (X,Y)	USR(X),Y
PAINT X,Y,MODE	SYS 49240,X,Y,MODE

Fig. 1

Computer-Kunst of kunde

We hebben deze aflevering wel zo resoluut de ondertitel ComputerKunst meegegeven, maar de vraag rijst natuurlijk onmiddelijk of het maken van computergraphics wel onder de naam Kunst mag worden gepresenteerd. Sommige mensen vinden computercreaties wel degelijk kunst met een grote K. Zij kunnen misschien hun creatieve gedachten juist door middel van de computer omzetten in beelden. We denken in dit geval bijvoorbeeld aan computeranimaties. Andere mensen vinden een computer juist creatief-beperkend werken. Ook het op het scherm doen verschijnen van, weliswaar vaak de prachigste figuren zoals fractals, verstaan zij niet onder kunst. Ook voor dit standpunt is wat te zeggen. Is het immers niet vaak zo, dat achter de verschillende kleuren en vormen gewoon wiskundige formules schuil gaan? In dit geval kan dan misschien beter van ComputerKunde gesproken worden. Wel, deze discussie is haar einde nog niet genaderd. Wij zullen u verder geen mening opdringen over het al dan niet kunst zijn van computercreaties. In ieder geval, met de volgende omschrijving van kunst: Kunst omvat alle dingen die door mensen gemaakt zijn en bovendien mooi zijn, vinden wij dat de ondertitel het gebodene dekt. Maar let wel, de



bovenstaande omschrijving van kunst is zuiver subjektief. Omdat het makkelijk praat, zullen we het verder hebben over Computerkunst.
Computerkunst omvat natuurlijk een heleboel dingen. Veel van deze dingen of onderdelen zullen wij niet behandelen, simpelweg omdat de Commodore 64 daarvoor niet rekenkrachtig genoeg is. Denk hierbij bijvoorbeeld aan complexe animaties. Omdat computerkunst zoveel omvattend is, zullen we de verschillende onderdelen verdelen over een paar afleveringen. Er komt bijvoorbeeld nog een aflevering over fractals. In deze aflevering zullen we het o.a. hebben over Lissajous figuren, sinuskrommen en de boom van Pythagoras.
Alle programma's zijn overigens geschreven in Simons's Basic. Mocht u deze of een soortgelijke uitbreiding niet hebben, dan kunt u het applikatie programma gebruiken dat in No 6 bij "Graphics op de 64" staat afgedrukt. Met behulp van de tabel in figuur 1 kunt u alle listings eenvoudig omzetten, zodat u ze samen met de Hires Routines kunt gebruiken. In de eerste kolom van deze tabel staan de Simon's Basic instructies met hun parameters. Bij elk van deze instructies staat in de tweede kolom in welke instructie(s) u de Simon's Basic instructie moet veranderen. In alle listings moet u voor {pi} gewoon de Griekse letter gebruiken.

Krommen

We zullen ons eerst gaan bezig houden met wiskundige krommen. Wiskundige krommen kunnen we op veel manieren verkrijgen. Door het werken met poolcoordinaten kunnen we bijvoorbeeld betrekkelijk makkelijk cirkels en spiralen tekenen. We beperken ons in dit artikel echter tot krommen waarvan de funktie in parametervorm is opgesteld. Dit houdt in dat de x- en y-coordinaat als funktie van een derde parameter, bijvoorbeeld t, wordt uitgedrukt. De x- en y-coordinaat zijn dan de gewone coordinaten op het beeld. (0,0) Linksboven en (319,199) rechtsonder. t is verder een variabele die tussen twee vastgestelde grenzen loopt. Door t steeds te varieren, worden steeds andere coordinaten gevonden. Vorige aflevering hebben we eigenlijk al een voorbeeld van een parametervorm gezien bij het tekenen van een cirkel. Toen hadden we de volgende funktie:

x = r * cos(t)
y = r * sin(t)

Hierbij is r de straal van de cirkel in pixels. Door in de onderste vergelijking voor y een andere straal te kiezen dan in de bovenste vergelijking, kan een ellips verkregen worden. De variabele t laten we lopen van 0 tot 360. t Stelt dan het aantal graden voor. Het programma voor het tekenen van een cirkel hebben we u de vorige aflevering al gegeven, dus dat doen we nu niet weer.

Lissajous figuren

We hebben een zeer eenvoudige parametervorm gezien in de vorige paragraaf. We gaan het nu hebben over funkties met de volgende parametervorm:

x = a1*sin(f1*t+v1) + a2*cos(f2*t+v2)
y = a3*sin(f3*t+v3) + a4*cos(f4*t+v4)

Deze parametervorm lijkt op het eerste gezicht tamelijk ingewikkeld. De x en y worden allebij bepaald door een sinus en een cosinus. a1 tot en met a4 zijn de amplitudes. f1 tot en met f4 bepalen de frequentie en v1 tot en met v4 bepalen de verschuiving van de sinus of cosinus. Figuren die beschreven worden door bovenstaande formule, worden Lissajous figuren genoemd. Deze figuren zijn ontdekt door de Franse fysicus Jules-Antoine Lissajous (1822 - 1880). We hebben in de inleiding gezegd dat we niet te diep op de wiskundige formules achter de verschillende figuren in zullen gaan. Dat doen we dan ook niet. We laten de formule zoals ze is, en zullen nu naar het programma gaan kijken die de Lissajous figuren op het beeld tekenen. Het volgende programma in Simons's Basic tekent willekeurige Lissajous figuren op het beeld:

```
5   rem Lissajous figuren
10  print"geef a1, a2, a3 en
    a4";
20  input a1,a2,a3,a4
30  print"geef f1, f2, f3 en
    f4";
40  input f1,f2,f3,f4
50  print"geef v1, v2, v3 en
    v4";
60  input v1,v2,v3,v4
70  hires 0,1
100 for gr=0 to 360
110 t=gr*{pi}/180
120 x=a1*sin(f1*t+v1)+a2*cos(
    f2*t+v2)
```

Waarden voor programma 1: Lissajous figuren											
a1	a2	a3	a4	f1	f2	f3	f4	v1	v2	v3	v4
100	0	100	0	9	0	10	0	0	0	45	0
40	60	40	60	6	5	5	6	0	1	1	0
30	70	30	70	6	5	5	6	0	0	0	0
10	90	50	50	2	4	8	16	2	3	4	6

Progr. 2: Sinuskrommen					
a1	a2	a3	f1	f2	f3
10	10	10	4	1	1
25	4	4	2	4	8
20	5	5	8	9	1
12	8	8	8	1	1
100	1	1	4	6	6

Progr. 3: Dr. lijnen		
le	dh	tl
5	144	3
4	70	2
4	125	2
3	92	1
100	145	0
100	72	-3
3	258	1

Fig.2

```
130 y=a3*sin(f3*t+v3)+a4*cos(
    f4*t+v4)
140 x=int(x+160)
150 y=int(y+100)
160 if gr=0 then x1=x:y1=y
170 line x1,y1,x,y,1
180 x1=x:y1=y:next
190 goto 190
```

In de eerste regels vanaf regel 10 moet u eerst de verschillende variabelen invoe-

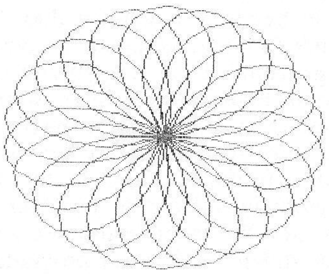


Fig. 3

ren. In regel 70 wordt het hires aangezet met HIRES 0,1. In regel 100 wordt een FOR-NEXT loop gedefinieerd van 0 tot 360. U kunt dit beschouwen als de variabele GR die van 0 tot 360 graden loopt. Omdat de computer niet met graden maar met radialen rekent, moet dit worden omgerekend. Dit gebeurt in regel 110. Daarna worden de x- en y-coordinaat volgens de gegeven parametervorm en de ingegeven variabelen berekend. Bij deze coordinaten tellen we nog de coordinaten van het midden van het scherm op, zodat de figuur mooi in het midden van het scherm komt te staan. Dit gebeurt in regel 140 en 150. Als laatste zou u natuurlijk verwachten dat er een pixel op het scherm werd aangezet. We trekken echter een lijn van de berekende coordinaten naar de coordinaten die in de vorige iteratie zijn berekend. Dit omdat er dan geen gaten in de figuur vallen. Bij de eerste iteratie zijn er echter geen vorige coordinaten berekend. Deze uitzondering wordt opgevangen met een IF-THEN in regel 160.
Het programma bevat echter een fout. Als de x- of y-coordinaat buiten het beeld valt, dan stopt het programma met een foutmelding als de lijn wordt getekend. Op deze situatie valt natuurlijk gemakkelijk te testen met een IF-THEN. De reden dat we dit niet hebben gedaan is dat het programma dan een stuk langzamer gaat lopen. Een manier om dit te voorkomen is om bij het invoeren van de a variabelen de volgende regels in acht te nemen: (a1 + a2) 160 en (a3 +a4) 100.
Als laatste wilt u natuurlijk weten met welke waarden voor de a, f en v variabelen u mooie figuren krijgt. In figuur 2 staat een tabel met waarden. Door deze waarden in te typen krijgt u mooie Lissajous figuren op het scherm. Wij raden u aan om verder zelf te gaan experimenteren met de waarden. De uitkomsten zullen vaak verrassend zijn.

Sinus krommen

We gaan een stapje verder, uitgaande van de parametervorm voor het maken van Lissajous figuren. Bekijk eens de volgende parametervorm:

x = r * a2 * sin(f2 * t)
y = r * a3 * cos(f3 * t)
r = a1 * sin(f1 * t)

U ziet dat we een extra variabele hebben geïntroduceerd, namelijk r. Dit is wederom een sinus. De x- en y-coordinaat worden met deze r vermenigvuldigd. De r is als het ware een soort variabele amplitude voor de x- en y-coordinaat. Met bovenstaande parametervorm kunnen we sinus bloemen en dergelijke tekenen. Het programma voor het implementeren van de parametervorm loopt voor het grootste deel parallel met het vorige programma. Hier komt het programma:

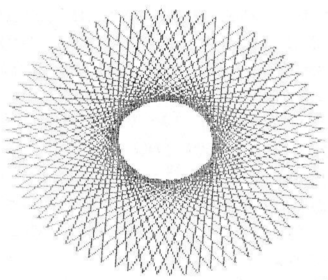


Fig. 4a

```
5   rem sinus krommen
10  print"geef a1, a2 en a3";
20  input a1,a2,a3
30  print"geef f1, f2 en f3";
40  input f1,f2,f3
70  hires 0,1
100 for gr=0 to 360
110 t=gr*{pi}/180
115 r=a1*sin(f1*t)
120 x=r*a2*sin(f2*t)
130 y=r*a3*cos(f3*t)
140 x=int(x+160)
150 y=int(y+100)
160 if gr=0 then x1=x:y1=y
170 line x1,y1,x,y,1
180 x1=x:y1=y:next
190 goto 190
```

Ook bij dit programma kan de error voorkomen die we bij het vorige programma hebben beschreven. De gebruiker kan wederom de fout voorkomen dat de x- of y-coordinaat buiten het beeld valt. Daarvoor moet de gebruiker de volgende regels in acht nemen bij het invoeren van de variabelen: (a1 * a2) 160 en (a1 * a3) 100.

U heeft inmiddels wel opgemerkt dat er geen v variabele in de parametervorm zit. Deze v was voor de verschuiving van de sinus of cosinus. U kunt zelf het programma aanpassen om deze variabele in te brengen. In figuur 2 staan waarden die u kunt uitproberen voor dit programma.
We hebben voor de variabele r nu een standaard sinus genomen. Er zijn ook andere leuke funkties te nemen voor r, om nog leukere figuren te maken. Als voorbeeld noemen we r = a1 * abs(sin(f1 * t)) + 50. Ook leuk is r = a1 * abs(cos(2 * sin(f1 * t))). Ook hiermee kunt u naar harte lust experimenteren. In figuur 3 staat een sinus kromme die met het bovenstaande programma is getekend. De waarden voor a1, a2, a3, f1, f2 en f3 zijn respectievelijk 100, 1, 1, 11, 6 en 6.

Draaiende lijnen

We gaan nu iets anders bekijken. We gaan lijnen trekken aan de hand van wat ingevoerde variabelen. De eerste variabele le definieert de lengte van de eerste lijn die getrokken wordt. De variabele tl definieert de lengte toe- of afname van de lijnen die worden getrokken. We geven een voorbeeld. De lenge van de eerste lijn is bijvoorbeeld le=5. Als tl nu 4 is dan wordt de tweede lijn getekend met lengte 5+4=9, de derde lijn heeft lengte 9+4=13 enz. Als derde variabele nemen we dh die de draaihoek van de te tekenen lijnen definieert. We gaan nog even door op het gegeven voorbeeld, met daarbij dh=45. Voordat we precies beschrijven wat er gebeurt op het scherm, moeten we nog twee begin constanten definieren. We nemen als startpunt van de eerste lijn het punt (160, 100). Als beginhoek van de eerste lijn nemen we 90 graden. De eerste lijn heeft dus als beginpunt (160, 100) en loopt onder 90 graden t.o.v. de x-as. Het eindpunt is dan te berekenen als de lengte

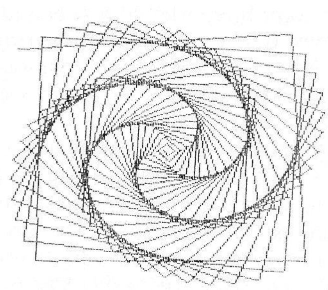


Fig. 4b

van de lijn bekend is (le). Het eindpunt nemen we als beginpunt van de tweede lijn. de lengte le wordt verhoogd met tl. We nemen nu als hoek 90+dh=135. Het eindpunt van de tweede lijn kan nu berekend worden en de lijn wordt vervolgens getekend. De derde lijn wordt getekend onder een hoek 135+dh= 180. Dit gaat zo door totdat de x- of y-coordinaat van een berekend eindpunt buiten het beeld valt. Het volgende programma implementeert het gegeven algoritme.

```
5   rem draaiende lijnen
```

```
10 input"geef lengte
   lijnstuk";le
20 input"geef draaihoek";dh
30 input"geef toename
   lijnstuk";tl
70 ho=90:hires 0,1
100 t=ho*{pi}/180
110 x=int(le*sin(t)+160)
120 y=int(le*cos(t)+100)
130 if x or x319 or y or y199
    then 190
140 if ho=90 then x1=x:y1=y
150 line x1,y1,x,y,1
160 x1=x:y1=y
170 ho=ho+dh:le=le+tl
180 goto 100
190 goto 190
```

U ziet dat het bovenstaande programma op zich redelijk makkelijk in elkaar zit. De coordinaten van het eindpunt van de lijn worden steeds berekend met een sinus en een cosinus. Met het bovenstaande programma zijn echt leuke figuren te maken. Ook voor dit programma staan wat waarden in figuur 1. Figuur 4 laat een figuur zien die met dit programma getekend is.

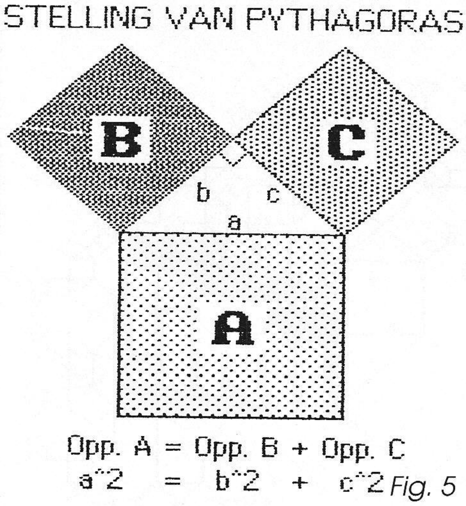


Fig. 5

Lissajous slang

De tot nu besproken vormen van computerkunst waren statisch van aard. De computer leent zich echter ook zeer goed voor het maken van bewegende kunst. Met behulp van de eerder besproken Lissajous figuren gaan we nu bewegende kunst maken.

Bij het tekenen van Lissajous figuren zult u wel gezien hebben dat vanuit een bepaald punt een vloeiende lijn wordt getrokken en dat deze lijn tenslotte eindigt in het beginpunt. We gaan nu echter geen lijn tekenen door de berekende coordinaten, maar laten acht sprites langs deze coordinaten bewegen. We laten de eerste sprite vertrekken vanuit het beginpunt; als deze sprite een paar stappen heeft gedaan dan laten we een volgende sprite vertrekken. Dit doen we tot alle acht sprites op het scherm staan. Er beweegt nu een 'slang' van sprites langs een mooie baan over het scherm.

Omdat Basic niet snel genoeg is, hebben we een machinetaal programma gemaakt dat de Lissajous 'slang' over het scherm laat glijden. De listing is listing 1 (aan het eind van het artikel). Deze listing moet u eerst intikken, saven en runnen. Het programma heeft voor zijn berekeningen twee tabellen met coordinaten nodig. Een tabel met alle x-coordinaten en een met

LISTING 1

```
 4 rem ********************
5 rem * lissajous slang *
6 rem *******************
7 rem
10 for i= 49152  to 49532
20 read a:s=s+a:poke i,a:next
30 if s53010 then print"fout in data!"
400 data 32,116,193,140,128,193,141,129
410 data 193,32,116,193,140,130,193,141
420 data 131,193,32,116,193,140,125,193
430 data 141,126,193,32,116,193,140,127
440 data 193,169,0,141,32,208,141,33
450 data 208,141,132,193,141,133,193,141
460 data 134,193,141,135,193,32,68,229
470 data 169,255,162,63,157,64,3,202
480 data 16,250,162,255,141,21,208,162
490 data 7,24,169,13,157,248,7,138
500 data 105,2,157,39,208,202,16,242
510 data 173,133,193,72,173,132,193,72
520 data 173,135,193,72,173,134,193,72
530 data 162,14,173,18,208,201,50,176
540 data 249,24,173,132,193,105,0,133
550 data 251,173,133,193,105,112,133,252
560 data 160,0,177,251,24,105,95,157
570 data 0,208,24,173,134,193,105,0
580 data 133,251,173,135,193,105,128,133
590 data 252,177,251,24,105,70,157,1
600 data 208,173,132,193,24,109,125,193
610 data 141,132,193,173,133,193,109,126
620 data 193,141,133,193,173,133,193,205
630 data 129,193,144,29,208,8,173,132
640 data 193,205,128,193,144,19,56,173
650 data 132,193,237,128,193,141,132,193
660 data 173,133,193,237,129,193,141,133
670 data 193,173,134,193,24,109,125,193
680 data 141,134,193,173,135,193,109,126
690 data 193,141,135,193,173,135,193,205
700 data 131,193,144,29,208,8,173,134
710 data 193,205,130,193,144,19,56,173
720 data 134,193,237,130,193,141,134,193
730 data 173,135,193,237,131,193,141,135
740 data 193,202,202,48,3,76,106,192
750 data 104,24,105,1,141,134,193,104
760 data 105,0,141,135,193,205,131,193
770 data 208,16,173,130,193,205,134,193
780 data 208,8,169,0,141,134,193,141
790 data 135,193,104,24,105,1,141,132
800 data 193,104,105,0,141,133,193,205
810 data 129,193,208,16,173,128,193,205
820 data 132,193,208,8,169,0,141,132
830 data 193,141,133,193,173,127,193,133
840 data 2,198,2,208,252,165,197,201
850 data 60,240,3,76,88,192,169,0
860 data 141,21,208,96,32,253,174,32
870 data 158,173,76,247,183
```

LISTING 2

```
4 rem ************************
5 rem * bereken coordinaten   *
6 rem * voor lissajous-slang *
7 rem ************************
8 rem
10 def fnx(t)=sin(2*t)*cos(5*t)
20 def fny(t)=sin(t)*cos(3*t)
30 sx=0:for i=0 to 2*{pi} step {pi}/800
40 poke 28672+sx,fnx(i)*80+80:sx=sx+1
50 next
60 sy=0:for i=0 to 2*{pi} step {pi}/800
70 poke 32768+sy,fny(i)*80+80:sy=sy+1
80 next
90 print "xstap =";sx
100 print "ystap =";sy
```

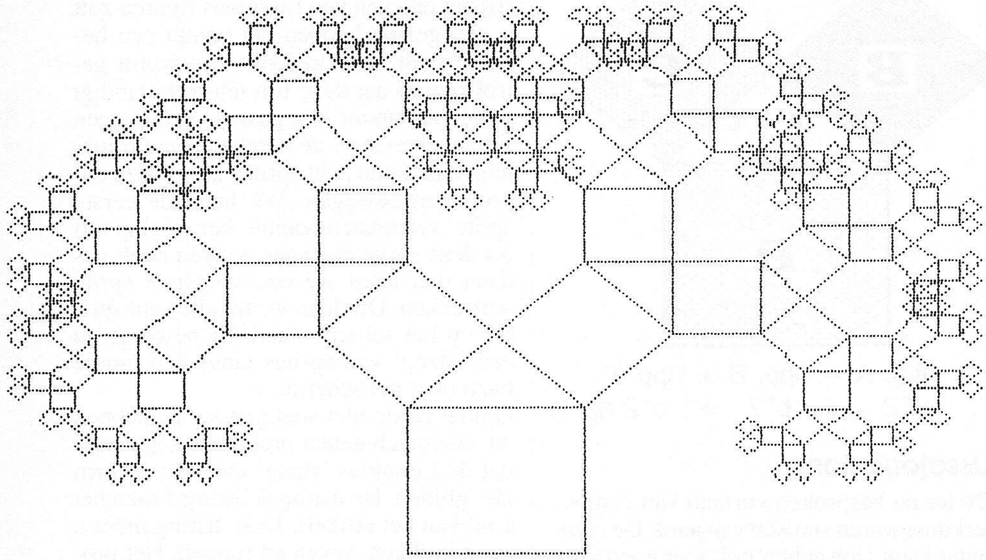


Fig. 6

y-coordinaten van de Lissajous figuur waarlangs de sprites zich voortbewegen. De eerste tabel begint op adres 28672 en de tweede op 32768. De toegestane x- en y-coordinaten zijn waarden tussen 0 en 255. Om nu een Lissajous slang te maken, moet u eerst de twee tabellen vullen. Dit kunt u met listing 2 doen. In regel 10 en 20 staan de functies voor de x- en y-coordinaat. Hier kunt u eventueel zelf een andere functie invullen. In regels 20 tot 50 worden alle x-coordinaten berekend en in regels 60-80 alle y-coordinaten. De beide FOR-NEXT lussen hebben een STEP van pi/800. Deze waarde kunt u eventueel veranderen. Hoe groter de waarde, hoe sneller de slang zal bewegen; de beweging wordt dan wel minder vloeiend. Het programma drukt na enkele minuten twee waarden af, t.w. xstap en ystap. Listing 2 retourneert de waarden 1601 en 1601. Nu kunt u het machinetaal programma starten. Dit doet u met de volgende instructie:

SYS 49152,xstap,ystap,tussen ruimte,snelheid

Voor xstap en ystap moeten de waarden, die de computer u gegeven heeft, worden ingevuld. tussenruimte bepaalt hoeveel ruimte er tussen twee opvolgende sprites in de slang zit (hoe groter de waarde van tussenruimte, hoe verder sprites van elkaar zitten). snelheid bepaalt hoe snel de slang beweegt (1=snel, 255=langzaam). Om nu de slang langs de baan, die door listing 2 berekend is, te bewegen, tikt u SYS 49152,1601,1601,10,1 in. Als het goed is beweegt de slang nu over het scherm. U kunt de slang stoppen door op de spatiebalk te drukken. Met de waarde voor de tussenruimte kunnen we leuke effecten krijgen. Probeert u maar eens 792 i.p.v. 10 voor de tussenruimte.

Om een andere baan van de slang te krijgen, moeten we regels 10 en 20 veranderen. Verandert u deze maar eens in:

10 def fnx(t)=sin(10*t+{pi}/3)
20 def fny(t)=sin(9*t)

Run nu het programma en tik daarna SYS 49152,1601,1601,10,200 in. De slang beweegt zich nu anders dan in het eerste voorbeeld. Een aantal leuke waarden voor tussenruimte zijn: 42, 83 en 166

Boom van Pythagoras

Ongeveer 500 v. Chr. formuleerde de beroemde Griekse wiskundige, Pythagoras, de stelling, die nu bekend staat als 'de stelling van Pythagoras'. Deze stelling zegt, dat in een rechthoekige driehoek het kwadraat van de schuine zijde gelijk is aan de som van de kwadraten van de andere twee zijden. Als we de schuine zijde nu aanduiden met de letter 'a' en de twee overige zijden met de letters 'b' en 'c' dan ziet de stelling van Pythagoras er in wiskundige notatie als volgt uit: a~2 = b~2 + c~2. In deze vorm zult u de stelling waarschijnlijk op de middelbare school geleerd hebben. (Had u toen kunnen denken dat u ooit nog eens iets leuks met de stelling van Pythagoras zou doen?). Pythagoras formuleerde zijn stelling echter op een iets andere wijze. We kunnen dit het best aan de hand van een plaatje laten zien. In figuur 5 staat een rechthoekige driehoek, met zijden a, b en c. Op elk van deze zijden is een vierkant getekend. Pyhtagoras formuleerde de stelling als volgt: de oppervlakte van vierkant A is gelijk aan de oppervlakte van vierkant B plus de oppervlakte van vierkant C.

Figuur 5 is dus de grafische voorstelling van de stelling van Pythagoras. Met deze grafische voorstelling gaan we nu een leuke figuur tekenen. Op de vierkanten B en C tekenen we een rechthoekige driehoek. Op deze driehoek tekenen we vervolgens twee vierkanten en op elk van deze vierkanten tekenen we dan weer een driehoek. Als we zo een tijdje doorgaan, ontstaat het figuur dat in figuur 6 getekend is. Deze figuur lijkt een beetje op een boom en wordt daarom 'de boom van Pythagoras' genoemd. Deze boom zou u natuurlijk gewoon met een potlood en een lineaal op papier kunnen tekenen. Dit is echter een tijdrovend karwei, dat met behulp van de computer veel sneller gedaan kan worden. In figuur 6 hebben alle driehoeken steeds twee basishoeken van 45 graden; hierdoor ontstaat een symetrische boom. (de basishoeken zijn de hoeken die grenzen aan de schuine zijde). Door niet twee hoeken van 45 graden, maar bijvoorbeeld een hoek van 60 en van 30 graden te nemen, ontstaat een scheef gegroeide boom. Het algoritme om de boom van pythagoras op de computer te tekenen is helaas nogal ingewikkeld. We geven daarom zonder uitleg de Simon's Basic listing van het programma om bomen van Pythagoras te tekenen.

```
20    rem * boom van pythagoras
      *
50    input "grootte van linker
      basishoek";al
60    input "hoeveel
      iteraties";m
70    al=al/180*{pi}
80    input "lengte van
      vierkant";l
90    input "coordinaten van
      linker basishoek";x1,y1
100   x2=x1+l:y2=y1:h={pi}/2
110   hires 0,15
120   gosub 220
130   poke 198,0:wait 198,1
140   end
150   rem *** teken vierkant
160   line x1,y1,x2,y2,1
170   line x1,y1,x3,y3,1
180   line x2,y2,x4,y4,1
190   line x3,y3,x4,y4,1
200   return
210   rem *** teken boom
220   i=i+1
230   x1(i)=x1:y1(i)=y1:x2(i)=
      x2:y2(i)=y2:h(i)=h
240   gosub 350:gosub 160
250   if i=m then 320
260   x1=x3:y1=y3:x2=x5:y2=y5:
      xd(i)=x4:yd(i)=y4
270   h=h+al:if h2*{pi} then
      h=h-2*{pi}
280   gosub 220
290   x1=x2:y1=y2:x2=xd(i):y2=y
      d(i)
300   h=h(i)-({pi}/2-al):if
      h2*{pi} then h=h+2*{pi}
310   gosub 220
320   x1=x1(i):y1=y1(i):x2=x2(
      i):y2=y2(i)
```

```
330     i=i-1
340     return
350     l=sqr((x1-x2)~2+(y1-y2)~
        2)
360     x3=x1+l*cos(h):y3=y1-l*
        sin(h)
370     x4=x2+l*cos(h):y4=y2-l*s
        in(h)
380     x5=x3+l*cos(al)*cos(al+
        h-{pi}/2)
390     y5=y3-l*cos(al)*sin(al+
        h-{pi}/2)
400     return
```

Het programma vraagt eerst om de grootte van de linker basishoek. Deze grootte moet in graden worden opgegeven en moet tussen 0 en 90 graden liggen. De grootte van de rechterbasishoek is dan automatisch 90-grootte.
Als tweede vraagt de computer om het aantal iteraties. Dit getal bepaalt hoe groot de boom wordt. In de eerste iteratie wordt het onderste vierkant (de stam) van de boom getekend. Tijdens de tweede iteratie wordt op dit vierkant een driehoek met daarop twee vierkanten gezet. In de derde iteratie wordt op elk van deze twee vierkanten weer een driehoek met twee vierkanten getekend. Tijdens elke iteratie groeit de boom dus een beetje.

Als derde vraagt de computer om de lengte van de zijden het onderste vierkant van de boom.
En als laatste moet u de coordinaten van de linkeronder hoek van dit vierkant invoeren.
In onderstaande tabel staan twee voorbeelden die u kunt invoeren. Als u de getallen uit de eerste kolom invult, dan krijgt u de boom uit figuur 6 op het scherm. De getallen uit de tweede kolom zorgen voor een scheef gegroeide boom.

	Figuur 6	scheef
Hoekgrootte	45	60
Aantal iteraties	8	7
Lengte vierkant	50	40
Coordinaten	135,199	80,199

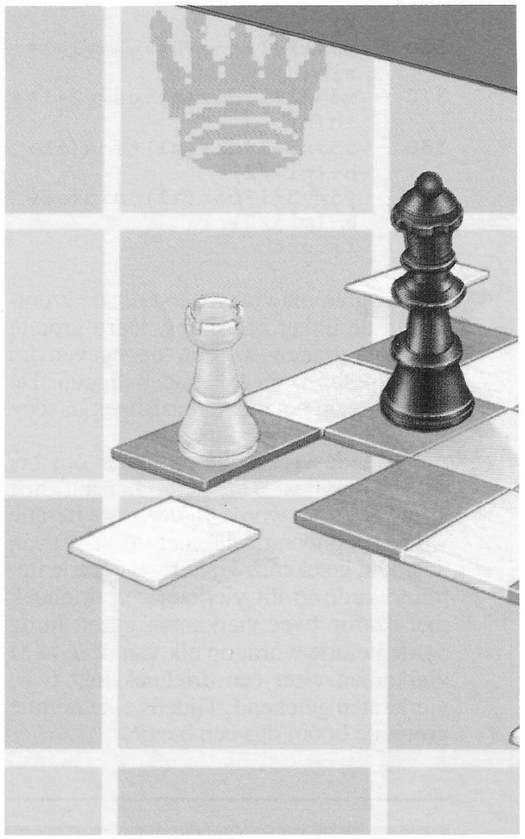
Als u een andere hoekgrootte probeert, dan moet u wat verschillende getallen voor de coordinaten en de lengte uitproberen, totdat de boom precies op het scherm past. Ter verbetering van het programma kunt u in de subroutine om het vierkant te tekenen (regel 160-190) testen of de coordinaten van de hoekpunten van het vierkant niet buiten het scherm liggen. Als dit het geval is verlaat u de routine meteen met RETURN. Hierdoor zal de computer niet stoppen met het tekenen van de boom, als er slechts een klein vierkantje buiten het scherm valt.
U kunt in plaats van vierkanten ook andere figuren tekenen. Daarvoor moeten regels 160-190 veranderd worden. In deze regels zijn (x1,y1), (x2,y2), (x3,y3) en (x4,y4) de coordinaten van de hoeken van het vierkant. Een eenvoudige verandering ontstaat door regels 160 en 190 te verwijderen. Daardoor worden maar twee zijden van het vierkant getekend, zodat de boom iets realistischer wordt. Een andere suggestie is, om i.p.v. vierkanten cirkels te tekenen. U moet daarvoor het middelpunt van het vierkant berekenen en om dit punt een cirkel tekenen. Laat uw fantasie de vrije loop en probeer maar eens wat leuke figuren.

Voor zover deel 6 over ComputerKunst. In de volgende aflevering zullen we het onder andere hebben over video banks en scrolling. Omdat er nog veel meer over ComputerKunst is te vertellen, zullen we in latere afleveringen nog terugkomen op dit onderwerp. Suggesties hier omtrent zijn van harte welkom.

Michel de Boer & Hylke Sprangers

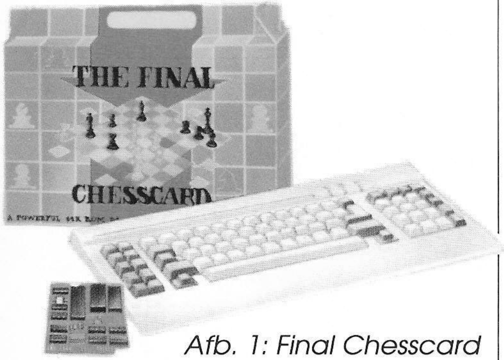
THE FINAL CHESSCARD

Een intelligente schaakcomputer



Wederom zijn de software-ontwikkelaars er in geslaagd iets nieuws te bedenken. Deze keer betreft het een cartridge van enige allure. Een denkspel voor de echte liefhebber, waarbij u uw computer verandert in een intelligente schaakcomputer.

Eerder al kwamen zij met een revolutionaire doorbraak op het gebied van cartridges, de Final Cartridge III. Nu komen zij, TASC genaamd, wederom met een cartridge voor de Commodore 64/128, de Final Chesscard, een schaakcomputer die volgens de advertenties menig andere schaakcomputer de baas is. We hebben deze cartridge eens aan een ander onderzoek onderworpen en zijn daarbij tot het volgende artikel gekomen. De cartridge wordt in een tweetal uitvoeringen uitgebracht, een Commodore en een PC versie. We beperken ons hier tot de Commodore versie. Was het eigenlijk wel nodig om een nieuw computerschaak programma op de markt te brengen? Eerder al verschenen voor de Commodore programma's als Chessmaster 2100 en Colossus Chess 4.0 op de markt. Deze programma's worden echter geleverd op diskette, hetgeen het schaken vertragen kan, omdat steeds informatie ingelezen moet worden.

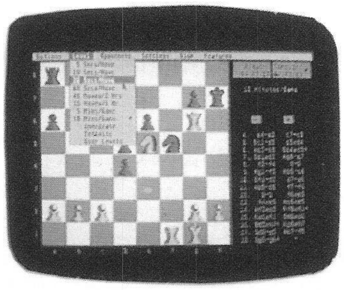


Afb. 1: Final Chesscard

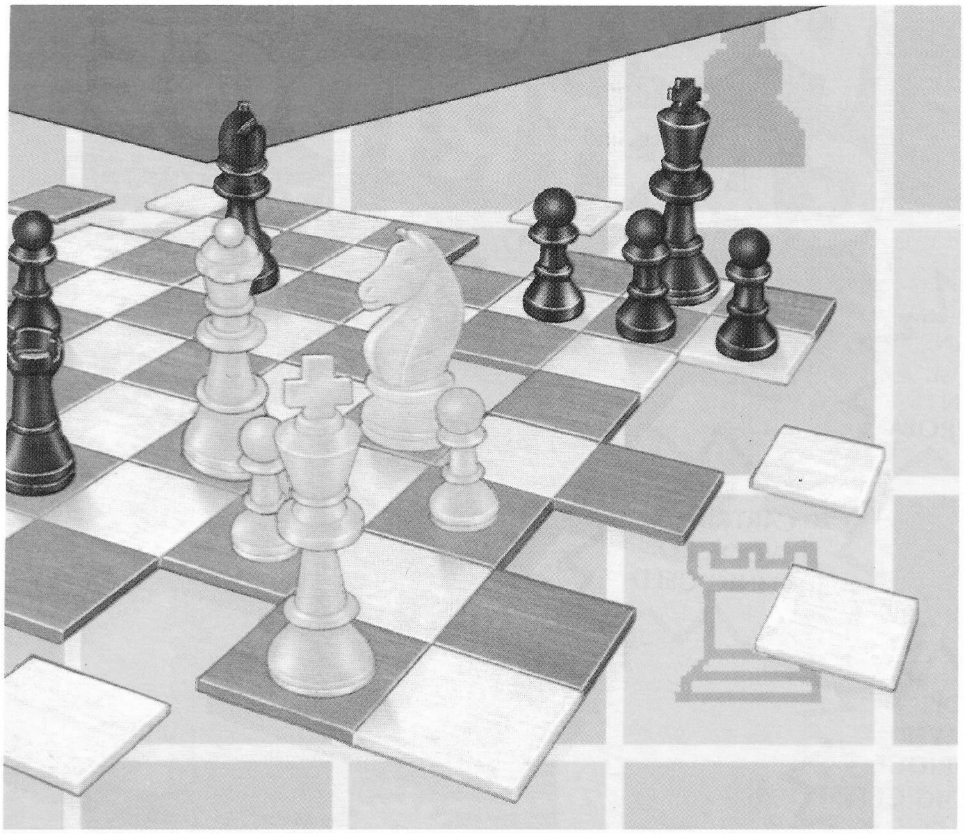
Met de komst van de Final Chesscard behoort dit storende euvel tot het verleden. Het programma wordt geleverd op een mooie compacte cartridge. Deze cartridge bevat een resetknop en biedt plaats aan een tweetal penlight batterijen. Hierdoor bestaat de mogelijkheid om informatie op te slaan binnen de cartridge en blijft informatie permanent intact.

Een eerste indruk

Als je de Final Chesscard aanschaft, de Commodore versie kost 199 gulden, dan krijgt men een cartridge met een duidelijke Nederlandse handleiding in een kleurige en stevige doos, zie afb. 1.
Je zou geneigd zijn de cartridge gelijk in de computer te steken en beginnen met spelen. Echter is het doorlezen van de handleiding aan te bevelen. Alhoewel de opzet van de Final Chesscard zeer gebruikersvriendelijk is, met pull down menu's, beschrijft de handleiding toch een aantal essentiële aspecten. Ten eerste wordt het installeren beschreven, waarbij het plaatsen van de twee 1,5 volt AA batterijen de meest ingrijpende ?? bezigheid is. Vervolgens wordt de cartridge in de expansiepoort van de C64/C128 geplaatst. U dient hierbij op te letten dat het apparaat uitgeschakeld is. De handleiding vervolgt dan met een introduktie over de verschillende mogelijkheden om stukken te verplaatsen. Hierbij heeft u de mogelijkheid uit verplaatsing met behulp van de muis, een joystick of het toetsenbord. Uiteraard is bediening met een muis of joystick te prefereren boven het gebruik van het toetsenbord. Het verplaatsen van een schaakstuk is op zich heel eenvoudig. Men klikt het aan door met de pointer (aanwijskruisje) op het te verplaatsen stuk te gaan staan en dan op de muisknop of vuurknop te drukken en deze ingedrukt te houden. Het schaakstuk zal nu van kleur veranderen en is verplaatsbaar. Wanneer u met het toetsenbord werkt is de functie van muisknop vastgelegd onder de CONTROL toets. Op deze manier verplaatst u het schaakstuk naar de door u gewenste positie. Een illegale verplaatsing, zoals bijvoorbeeld het schuin verplaatsen van een pion, wordt door de Final Chesscard genegeerd en het stuk komt terug op zijn oude positie. Op deze manier zijn verkeerde zetten niet mogelijk. Rechts in het scherm wordt de zgn. partijnotatie bijgehouden. Dit geschiedt geheel automatisch. Hierbij heeft u ook nog de mogelijkheid deze partijnotatie af te laten drukken



Afb. 2: Scherm afdruk



The Final Chesscard, een nieuw schaakspel voor de CBM 64/128 en iBM compatible machines, van de makers van de Final Cardridge

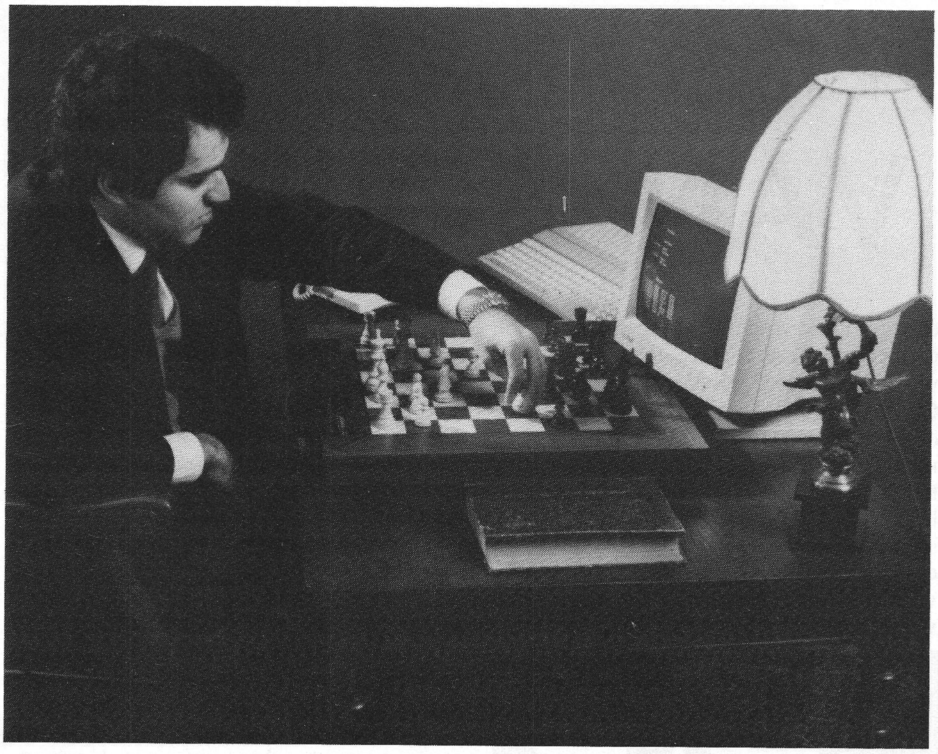
op de printer. Deze optie is vooral handig wanneer u een schaakpartij achteraf wilt analyseren, zie afb. 2.
U kunt eveneens gedane zetten terugnemen door op de neerwaartse pijl aan de rechterzijde van het scherm te klikken. Hierbij is tevens de mogelijkheid om teruggenomen zetten weer vooruit te zetten, hetgeen u realiseert met de opwaartse pijl. Twee computerklokken registreren de tijd van iedere deelnemer. Bovenin het scherm wordt eveneens de hoofdvariant afgebeeld. Deze bestaat uit maximaal 5 zetten, gevolgd door de bereikte zoekdiepte en een stellingswaardering. Deze waardering geeft aan welke speler het beste voorstaat.

Scherm lay-out

Zoals eerder reeds vermeld maakt de Final Chesscard gebruik van pull-down menu's, windows en gadgets. In een window wordt bepaalde informatie weergegeven, waarbinnen zich weer tekstgadgets bevinden. Ze worden tijdelijk over het speelveld heen geplaatst. In de gadgets, de kleine vakjes omgeven door een kader, kan met behulp van de pointer een opdracht worden geselecteerd of een waarde worden gewijzigd. Met de pull-down menu's kunnen diverse instelwaarden worden gewijzigd. Om u een indruk te geven van de instellingen die de Final Chesscard zoal heeft volgt hier een opsomming. Als eerste kennen we het options menu, hier binnen kan worden gekozen voor de start van een partij of een herstart. Met de optie move wordt de partij uiteindelijk geaktiveerd. De optie stop thinking dwingt de computer onmiddellijk te stoppen met denken en zijn tot op dat moment beste zet te doen. Een scala aan zgn. preset levels zijn mogelijk, bijvoorbeeld het aantal seconden per zet, het aantal zetten per 1 of meerdere uren. Een partij van 5 tot 10 minuten. Alle niet onder de preset level in te stellen waarden kunnen met de optie user level worden ingesteld. De play mode stelt u in staat op te geven aan de computer tegen wie u wilt spelen, dit kan zijn speler-speler, speler-computer of computer-computer. Voorts kan hier opgegeven worden of er een zet teruggenomen moet worden of dat alle zetten teruggezet moeten worden. De optie replay a move en replay all geven u respectievelijk de gelegenheid om een zet of alle zetten opnieuw te spelen.

Het speelbord

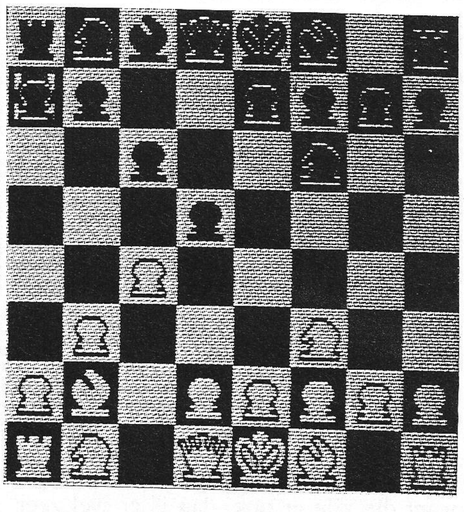
Ook ten aanzien van het schaakbord zijn er een aantal presets in te stellen. Zo kunt u kiezen voor de optie teaching, waarbij de Chesscard u alle velden laat zien waarop een legale zet mogelijk is. De optie suggest move laat u een mogelijke zet zien. Om de coördinaten op bij het bord te plaatsen kiest u de optie coordinates. Het bord is met de optie rotate board 180 graden te draaien, vooral handig als u met zwart speelt. De Chesscard kent drie snelheden, normaal, snel en onmiddellijk. Verder zijn de kleuren van zowel de menu's, de stukken als het scherm naar uw eigen smaak te veranderen. Met de optie settings kunnen een aantal instellingen gewijzigd worden die betrekking hebben op onder meer de informatie die de Chesscard laat zien, de tijdsvermelding, in tijd per zet of totaaltijd en betreffende de zgn. auto move, waarbij de computer pas gaat rekenen op het moment dat hij aan zet is. De Chesscard biedt ruime mogelijkheden om partijen te laden, te saven en te verwijderen op diskette. Verder is er een mogelijkheid om een openingsboek of een expansie te laden. Ook het laden en bewaren van instellingen (settings) behoort tot de mogelijkheden. Enkele speciale mogelijkheden zijn onder meer het instellen van de stellingen, het aktiveren van een matzoeker hetgeen inhoudt dat u een zoekdiepte in moet stellen. U kunt commentaar aan een partij toevoegen met de optie add comment. Met behulp van een printer kunnen de opties print board en print each move zeer behulpzaam zijn wanneer een partij naderhand geanalyseerd moet worden.



Helaas worden de stukken niet grafisch op een bord geprint, maar als nummers op hun positie afgedrukt. De optie print each move houdt in dat steeds na het zetten van een schaakstuk de printer de coördinaten afdrukt. De optie about the card geeft u informatie over de CROM (het versienummer van de interface) en de BROM (het versienummer van het schaakgedeelte). Dit is belangrijk voor later, daar de leverancier in een bijgaand schrijven het al heeft over toekomstige uitbreidingen, zoals printerutilities, een openingsboek editor etc.

Algemeen

In de bijgeleverde handleiding worden nogal veel schaaktermen gebruikt, weliswaar worden deze zeer helder uitgelegd. Meestal hebben deze termen betrekking op de uitgebreide mogelijkheden van dit stukje hardware. Begrippen als openingsboek, waarin de meest gebruikelijke openingszetten zijn opgeslagen. Of de term Brute Force, hetgeen betrekking heeft op het schaak-algoritme. Wanneer deze niet is geselecteerd zal de Chesscard met de Shannon B methode spelen (alleen relevante zetten worden bekeken). Is de Brute Force echter wel geselecteerd dan wordt gespeeld volgens de Shannon A methode, zodat alle mogelijke zetten worden bekeken. Verder bestaat de mogelijkheid om in de beginners modus te spelen, waarbij de Chesscard zijn speelstijl aan uw stijl aanpast. Dit betekent bijvoorbeeld dat als u een loper verliest de Chesscard na enige tijd ook een loper prijsgeeft. Ondanks het feit dat de Final Chesscard al zeer aktief speelt, kan met de optie aggresive playstyle het spelgedrag nog aktiever worden ingesteld. Als u eerst een demonstratie van de Chesscard wilt houdt u de INST/DEL toets ingedrukt terwijl u de computer inschakelt. Het spelen met de Final Chesscard is in een woord plezierig te noemen. De vele parameter instellingen maken het mogelijk de card naar uw eigen speelgedrag in te stellen. Zo is het dus ook mogelijk om als beginnend schaker met deze card uit de voeten te kunnen. Hetgeen ook het opleidingsinstituut NTI beoogt met het volgen van een cursus. Een interaktieve cursus met een beginnersniveau, samengesteld door twee vermaarde schaakmeesters C. van Wijgerden en R. Brunia. Hun lesmethode is gebaseerd op jarenlange K.N.S.B. ervaring. De cursus bestaat uit een 8-tal diskettes en is didaktisch zeer verantwoord en beproefd.



Karakteristieken

De Final Chesscard werkt met pull-down menu's waardoor het bedieningsgemak optimaal is. De besturing van het schaakbord kan worden gerealiseerd met behulp van een muis (1351), een joystick of het toetsenbord. De kleuren van het bord, de schaakstukken en het scherm zijn naar eigen keuze in te stellen. De speelsterkte voor de Commodore versie gaat tot maar liefst 2000 Elo en kan worden ingesteld op selective search of brute force. De bedenktijd is volledig instelbaar en wordt bijgehouden op de klokken, die de tijd registreren in totaaltijd, zettijd of countdown. De zoekdiepte is instelbaar met de mogelijkheid een matzoeker in te stellen die werkt tot 10 ply. Een teaching mode toont legale zetten, alsmede een suggest move optie maken dit pakket tot een educatief geheel. Terwijl u aan zet bent denkt de computer na, maar deze optie kan ook uitgeschakeld worden. Partijen kunnen worden opgeslagen op diskette, alsmede standaardinstellingen zoals kleur, tijd en speelsterkte. Voor de analyse van een gespeelde partij bestaat de mogelijkheid tot het afdrukken van een diagram. Uitbreidingsmogelijkheden met openings en trainingsdiskettes.

Conclusie

Met de komst van de Final Cartridge is het software-arsenaal voor de Commodore niet alleen uitgebreid met een gewoon schaakprogramma, maar men krijgt de beschikking over een sterke tegenstander en een groot medestander als sekondant, oefenmeester of trainer. Plezierig is het gebruik van een cartridge, wat het vele bijladen van mogelijkheden tot het verleden laat behoren. Hierdoor is de Final Chesscard snel en krachtig. Met andere woorden uw Commodore 64/128 wordt in feite met het plaatsen van de cartridge omgebouwd tot een heuse schaakcomputer. Kortom een brokje hard/software waar we in de toekomst best nog wel wat meer van zullen horen.

Beoordeling

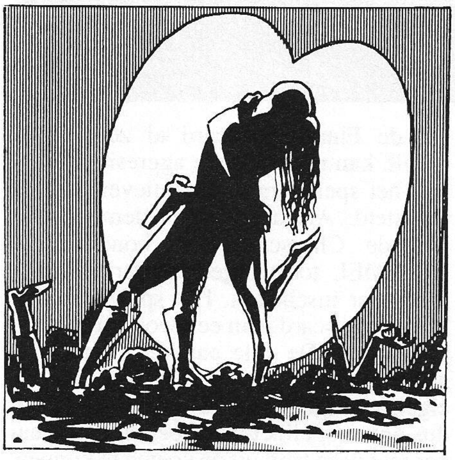
spelniveau	:	zeer goed
kleurgebruik	:	goed
kwaliteit	:	zeer goed
documentatie	:	goed

Bert Venema

OUD VAN GOUD

Rob Goudriaan trok weer een aantal ouwetjes uit de kast en besprak voor de liefhebbers hun tijdloze merites.

Een of andere plaaggeest heeft al het geld gestolen en heeft daarmee de gehele wereld in de war gestuurd. Een grappig uitziend mannetje moet dit zien te vinden daarmee redt hij de wereld van de ondergang. Dit springmannetje is een soort spring in het veldje, een spring mannetje die door zijn vriendelijk uiterlijk danig door zijn vijanden onderschat kan worden. De raarste kapriolen kan hij maken, geen berg of in dit geval, geen hindernis is hem te hoog door zijn veer kan hij hele hoge sprongen maken en dat dit nodig is zul je al gauw ontdekken. Inzicht en precisie zijn de ingredienten die nodig zijn om deze opdracht te voltooien. De vreemdste hindernissen kun je tegenkomen. Op diverse plaatsen zijn pakjes verstopt deze zijn nodig om verder te kunnen gaan anders kom je zeker hindernissen tegen die onmogelijk te nemen zijn. Een roltrap bijvoorbeeld kan je alleen nemen als je het bijpassende cadeautje hebt opgepakt. Je kan ook alleen van de lift gebruik maken als je ook dit bijbehorende cadeautje in je bezit hebt. En zo heeft iedere hindernis zijn eigen presentje en of dit nu voor het transporteren door de lucht is of voor het gebruik van een trampoline. Behalve de hindernissen waar voor de cadeautjes nodig zijn, zijn er ook nog andere gevaren en vergis je hierin niet of het nu vriendelijk ogende waggeltjes de happertjes uit pacman spuitbussen of telefoons zijn, ze vreten energie. Allerlei vreemde uitrustingen hebben deze energie-vreters de een ziet er wat gevaarlijker uit als de ander maar laat u niet misleiden zelfs niet door kussende mondjes. Kom je deze tegen, zeg dan niet dat we niet hebben gewaarschuwd. Kijk uit voor transportbanden deze gaan maar een kant op. Zo als al gezegd is er een lift. Heb je het bijpassende kadootje gevonden dan kan je hier mee naar de tweede, derde of vierde etage. Hier staan weer andere gevaren op je te wachten. Laser stralen kosten ongelofelijk veel energie, en ook hier moet er weer heel wat op en af gesprongen worden. Kijk uit dat je niet ergens door heen zakt want dit kan het einde betekenen. Omdat alles wat je doet en aanraakt energie vreet, staan of hangen er hier en daar energie tanks. Maak hier gebruik van want anders haal je het einde zeker niet. Op diverse plaatsen hangen er plaatjes, deze moeten verzameld worden, zij vormen straks een geheel. Kijk rechts boven, hier verschijnen de plaatjes die je gepakt hebt. Het zal niet direkt lukken want dit spel kost de nodige moeite, maar heb je eenmaal door hoe je moet springen en waar alles voor dient, dan is dit spel zeer verslavend, u bent gewaarschuwd.



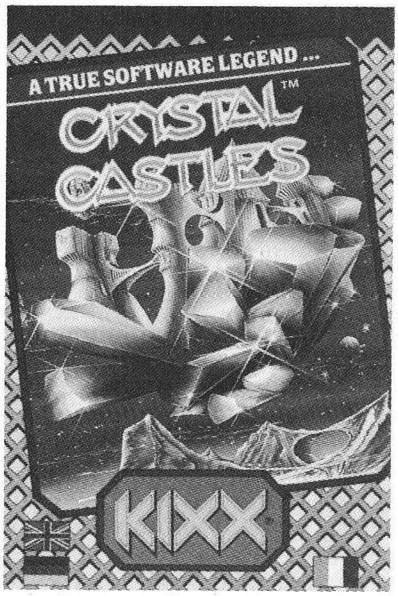
Hopeloos

Bij dit spel is het de bedoeling dat u op harten jaagt. Dit moet gedaan worden in de eindeloze ruimten van New Almere. Voorzien van een speciaal soort ruimte pak gaat u van start. U heeft een speciaal motortje op uw rug waarmee u door de oneindige ruimten kunt zweven, want de harten liggen nu niet bepaald op een hoopje. Er moet heel wat voor worden gedaan. In de dubieuze wereld van Manic Munk is alles mogelijk en je hebt dan ook aan hem een zeer geduchte tegenstander. Als u de benodigde harten hebt verzameld redt u de gekidnapte vriendin uit de handen van dit genadeloze wezen. Hoe vernuft dit wezen is zult u al gauw ontdekken. De harten zijn nu niet gemakkelijk te bereiken. Een paar zijn er redelijk te verkrijgen. Dit is om u nu niet al te zeer te *ontmoedigen maar laat u niet misleiden*, diversen zijn nep en heeft u nu net een helse strijd gevoerd tegen een zo'n talrijk leger van tegenstanders, muren en handels die om gezet moeten worden, dan blijkt het een waardeloos hart te zijn, want die zijn er ook, dan is er wel zeer veel doorzettings vermogen voor nodig om door te gaan. Op diverse plaatsen zijn monitoren geplaatst. Deze hebben verschillende functies. De meest voor komende zijn de overzicht monitors hier op kunt u zien waar u uithangt, de omgeving kunt u bekijken en bepalen hoe ver u nog moet en of u op de juiste weg zit. Een knipperend sterretje wijst u de "goede" weg, daar is een hart verborgen. Staat u voor de monitor met het knipperend hart, land hier door vuurknop indrukken handel naar voren en als alles goed is is er een heart attack en het eerste hart is in uw bezit. Ergens in het grootte heelal verdwijnt er een hindernis die een weg vrij moet maken naar het binnenste van het hart. Behalve deze twee monitors zijn er ook nog monitors waar de nodige brandstof bijgetankt kan worden. De makkelijkste zijn wel de transporteer monitoren met behulp van deze kan je een hele afstand verder terecht komen. Dan staan er in het begin van de diverse ruimte grotten monitoren die bepaalde schakelingen verrichten. Deze schakelen horizontale openingen in vertikale openingen al naar gelang dit nodig is. Dan zijn er ook nog monitoren met kleuren schakelingen, uiteraard is dit alleen voor de kleur die er op staat voor de rest moet u de bijbehorende kleuren monitor op zoeken. Let goed op het paneel rechtsonder, deze geeft diverse handige tips en informatie door, deze houd bij hoeveel harten er nog moeten worden verzameld en hoeveel mannetjes u nog heeft. Hoe het met de energie voorraad staat enz. Voor ieder hart dat u gekraakt heeft verdwijnt er een hindernis zijn alle harten, op een na gevonden, dan kunt het centrum der harten betreden. Hier moeten de laatste hindernissen genomen worden waaronder het laatste hart. Persoonlijk vind ik het einde wat teleurstellend na al die moeite om de harten te sparen had het einde wat anders gekund, maar dat is mijn persoonlijke mening toch is dit spel zeker de moeite waard om te spelen.

Crystal Castle

In het spel Crystal Castel kruip je in de huid van een beer, genaamd Bentley. Bentley heeft een goedaardig karakter en zal nooit een vlieg kwaad doen. Zijn belangrijkste aktiviteit is zo af en toe een

potje honing te eten en eens een dutje te doen, meer heeft hij niet nodig. Hierop is een grote uitzondering, het gems-virus. Zodra hij de gem kriebels krijgt is er geen houden meer aan. Gems zijn duidelijk te herkennen, ze liggen overal verspreid over de vloer. Er zijn maar liefst 18 verschillende locaties, allemaal in 3-dimensionale uitvoering. U kunt van alles op uw weg tegenkomen wat uw werk kan vergemakkelijken, maar vaker wordt uw weg



er duidelijk moeilijker door. Zo zijn de liften een uitkomst om ergens snel te komen, maar ook de tegenstanders maken dankbaar gebruik van deze route. De tegenstanders zullen op alle mogelijke manieren proberen te beletten dat de gems worden opgepakt. Behalve de liften, zijn er ook nog hellingen, tunnels en ook schuilplaatsen. Deze schuilplaatsen zijn een prima plaats om je te verstoppen voor hinderlijke achtervolgers. De tegenstanders laten zich eigenlijk niet beschrijven, ze zijn in staat steeds een andere gedaante aan te nemen, dit maakt de opdracht er niet gemakkelijker op. Hierdoor moet steeds goed opgelet worden, wie en waar is de vijand. Elke tegenstander heeft zijn eigen wapen, hun opdracht is echter voor een ieder gelijk, vernietig de goedaardige beer. Ook de laatste gem blijkt voor hun een gewilde prooi te zijn, deze heeft namelijk bonus waarde, en is daardoor extra interessant. De tegenstanders zijn even in verwarring te brengen door over ze heen te springen. We hoeven U niet te vertellen dat dit ook extra gevaren met zich mee brengt. Een stuur of denkfout is hierbij snel begaan, maar je moet er wat voor over hebben voor wat tijdwinst. Een moeilijke tegenstander is het skelet, deze doolt over de weg zonder een bepaald patroon. Hierdoor is dit een onberekenbare tegenstander. Berthida, dit is een gemene heks. Zij kan alleen worden vernietigd als je in het bezit bent van de magic-hat, deze maakt je voor een korte periode onzichtbaar. De hoed kan in je bezit komen door er over heen te lopen of te springen. Heel verstandig is het niet te lang op één plaats te blijven staan, dit trekt direkt een horde tegenstanders aan. Beter is het in zo'n geval even een blokje extra rond te lopen. Een pot honing levert, net als de magic-hat, een flink aantal bonuspunten op. Het vernietigen van een gem eter of Berthilde levert 1000 punten op. Ook de laatste gem per scherm kan de score flink doen oplopen. Veel succes met de tree wave, gems, Berthilda en alle andere vreemde wezens.

Scooby Doo

Vele mensen zullen Scooby Doo wel herkennen van de gelijknamige televisie serie. Nu heeft U dan de kans om in zijn huid te kruipen. Net als in de film is er maar één opdracht, het bevrijden van je vrienden en zelf proberen om uit handen te blijven van je tegenstanders. Velma, Daphne en Fred zijn gevangen genomen door een boosaardige professor. Op dit punt houd de overeenkomst tussen het computerspel en de televisie serie geheel op. De Scooby Doo in dit spelletje moet er heel wat op los boksen voor hij maar in de buurt van zijn vrienden kan komen. De professor heeft ze verstopt in een grote stopfles. Ze zijn slechts één voor één te bevrijden. Het speelveld is opgebouwd uit een groot aantal schermen. Deze scrollen, bewegen, als achtergrond over het beeld. Op het moment dat het spel wordt gestart staat U, als Scooby Doo links onder op het scherm. Tijdens het lopen naderen al snel de eerste tegenstanders. De grote zwevende geest is uit te schakelen door hem een ferme tik te geven. Niet alleen deze grote geesten maken het je lastig maar zo mogelijk nog vervelender zijn de kleine spookjes. Ze komen op de meest onverwachte plaatsen uit een deur op de achtergrond ineens opduiken. Niet alleen van voren dreigt er gevaar, dus let ook goed op wat er achter je gebeurt. Over schedels, deze suggereren eerdere tegenstanders die minder geluk hadden, kun je heen springen. Ook de gaten in de vloer zijn op deze manier te overwinnen. De trappen zijn er om beklommen te worden, maak er selektief gebruik van. Niet alle trappen leidden naar de goede weg. Zeker je moet niet gauw de moed verliezen, het is en blijft puzzelen in dit grote doolhof. Nu zou dit spel niet spannend zijn als de professor de vrienden niet goed verstopt zou hebben, we verraden dus niet waar ze zitten. We kunnen alleen verklappen, erg gemakkelijk maakt hij het je niet. Op het moment dat je de eerste vriend, of vriendin hebt gered, moet je niet denken nu weet ik de weg. De professor is gemeen en zeker niet dom. Net als je denkt nu weet ik de weg, blijkt het gehele doolhof andere gangen, trappen en ingangen te hebben. Er zijn onderweg gelukkig extra levens te verdienen. Dit kan gebeuren door de kleine schatkistjes, die onderweg zijn verstopt, te pakken. Deze zijn herkenbaar aan de letter "S". Kijk hier extra goed naar uit, ze zijn zeker noodzakelijk om dit spel goed uit te spelen. In het eerste level blijken in de praktijk de tegenstanders nog wel te overtroeven. Hoe verder in het spel, hoe moeilijker het geheel wordt. Er komen extra gaten in de vloer en er komt een heel springerig volkje om de hoek kijken. In het derde level komt er een groot soort walvis op de proppen. In het vierde en laatste level heb je te maken met een speciaal soort aap, of iets wat daar op moet lijken. Al met al is het zeker geen gemakkelijke klus om alle vrienden uit de stopfles te bevrijden. Maar het spel staat zeker garant voor vele uurtjes spel plezier.



Tips & Trucs 64



De kerstdagen zijn nog niet aangebroken en het nieuwe jaar is ook nog even wachten, maar toch vast onze wensen aan u. Dit is tenslotte het laatste nummer van het jaar: prettige kerstdagen en een gelukkig nieuwjaar.

In deze aflevering van Tips & Trucs zullen we het onder andere hebben over hoe Basic programma's worden opgeslagen en over het veranderen van regelnummers. Ook zal worden besproken hoe een zelf-runnend programma gemaakt kan worden.

Basic programma's

Zoals u waarschijnlijk wel weet, wordt er in de adressen 43 en 44 het beginadres van een Basic programma bijgehouden. Als er een programma in de computer zit, wordt door deze adressen het beginadres gegeven, waar het programma start. Als u begint met programmeren, wordt het programma ook opgeslagen vanaf dit adres. Het beginadres is als volgt te berekenen:

beginadres = peek(43) + 256*peek(44).

Het eindadres van het programma wordt bewaard in de adressen 45 en 46. Het eindadres berekenen gaat op een soortgelijke manier:

eindadres = peek(45) + 256*peek(46).

We noemen de adressen die zelf een adres bevatten pointers. De adressen 43 en 44 vormen zo samen een pointer naar het beginadres van een Basic programma.
We gaan nu kijken hoe een Basic programma wordt opgeslagen in het geheugen.
Als eerste moet er op het beginadres-1 een nul in het geheugen staan. Het beginadres staat standaard op 2049. Dit betekent dat er op adres 2048 een nul in het geheugen moet staan. U kunt bijvoorbeeld in adres 2048 een 1 poken. Als u dan een Basic programma probeert uit te voeren, zult u merken dat dit niet gaat. De computer geeft een foutmelding.
Verder is er ook bepaald dat er aan het eind van een Basic programma drie opeenvolgende nullen in het geheugen moeten staan. U kunt met een monitor het geheugen bekijken. U kunt het eind van een programma dan herkennen aan het feit dat er drie nullen achter elkaar staan. U kunt bijvoorbeeld de pointer die het eind van het Basic programma aangeeft (de adressen 45 en 46 dus) verhogen. Hierdoor kunt u een extra stuk geheugen meesaven. In dit extra stuk geheugen kan dan bijvoorbeeld een hiresplaatje staan of iets dergelijks. Door nu met een monitor in het geheugen te gaan graven, kunt u toch nog kijken waar het programma werkelijk eindigt, door naar drie nullen te speuren. Overigens kunt u de drie nullen ook wel gelijk opsporen met het commando hunt, dat in de meeste monitors wel aanwezig is. We gaan nu kijken hoe de Commodore 64 globaal een programmaregel in het geheugen opslaat. We doen dit aan de hand van een klein voorbeeld programma:

```
10 print"hallo"
20 print"Greetings from the ArtVark"
```

Het beginadres van het bovenstaande programma is gewoon 2049. Op dat adres begint dus regel 10. De regels in een Basic programma worden uiteraard gewoon achter elkaar opgeslagen in het geheugen. Deze regels beginnen elk met 4 bytes die informatie over de betreffende regel aan de computer geven. Achter deze 4 bytes staat de werkelijke code voor die regel. Daarbij zijn de Basic instrukties gekodeerd opgeslagen. De print instruktie heeft bijvoorbeeld als code het getal 153. We gaan nu wat dieper in op de 4 informatie bytes. De eerste twee bytes geven het adres waarop de volgende regel begint. De eerste twee informatie bytes van regel 10 in ons programma zijn de adressen 2049 en 2050. Deze vormen samen dus een pointer naar het beginadres van regel 20. In ons programma bevat adres 2049 de waarde 14 en adres 2050 de waarde 8. Regel 20 begint zo op adres 14 + 256*8 = 2062. Op adres 2062 staat dus het eerste informatie byte van regel 20.
We hebben het nu gehad over de eerste twee informatie bytes. De tweede twee informatie bytes vormen samen het regelnummer van de regel. In ons voorbeeld bevat adres 2051 de waarde 10 en adres 2052 de waarde 0. Deze waarden vormen samen het getal 10 + 0*256 = 10.
Deze vier informatie bytes zijn bijvoorbeeld bij GOSUB instrukties nodig. De computer begint dan vooraan te zoeken naar het opgegeven regelnummer. Klopt het eerste regelnummer niet met het opgegeven nummer, dan gaat de computer naar de volgende regel enz. Om de snelheid van uw computer op te voeren kunt u subroutines het beste vooraan het programma zetten.

Regelnummers in Basic.

In de paragraaf Basic programma's hebben we het gehad over informatie bytes die de computer voor elke regel op slaat. De tweede twee informatie bytes bevatten het regelnummer. We gaan nu de regelnummers verder bekijken. Een byte kan als grootste getal de waarde 255 bevatten. Het theoretisch grootste regelnummer is op deze manier 255 + 256*255 = 65535. Echter, dit is de theorie. In de praktijk blijkt dat u niet een regelnummer groter dan 63999 kan kiezen. Probeert u maar eens het volgende statement in te voeren:

64100 ?"42"

De Commodore zal hier eenvoudig op reageren met een syntax error, omdat hij zo'n hoog regelnummer niet toestaat. We weten echter inmiddels wel waar de computer de regelnummers opslaat. Op deze manier kunnen we nu betrekkelijk makkelijk regelnummers genereren met waarden hoger dan 63999. Typt u maar eens het volgende in:

10 print"hallo"
20 print"dag"

De adressen 2051 en 2052 bevatten nu samen het regelnummer van regel 10. Nu voeren we de volgende poke in:

poke 2052,255

We hebben nu het regelnummer van regel 10 verandert in regel 65290 (10 + 256*255 = 65290). Als u list intypt krijgt u het volgende:

65290 print"hallo"
20 print"dag"

Het programma kunt u overigens nog gewoon steeds runnen. We kunnen nu echter regel 20 makkelijk nog een keer invoeren:

20 print"dag"

Wat we nu krijgen is het volgende programma:

20 print"dag"
65290 print"hallo"
20 print"dag"

U ziet het, de computer 'ziet' niet meer dat er al een regel inzit met nummer 20. Dit komt omdat de computer van voor af aan begint te zoeken naar een regelnummer 20. Bij regelnummer 65290 stopt de computer omdat 65290 groter is dan 20. De computer verwacht dus geen kleiner nummer achter 65290. U kunt het bovenstaande programma overigens gewoon uitvoeren. Er is echter wel een geval waar in het fout kan gaan. Er kan niet meer naar de laatste regel gesprongen worden. Als er nu een GOTO 20 gegeven wordt, springt het programma naar de eerste regel 20. Als deze niet zou bestaan, dan vindt de computer niet de achterste regel 20. De computer stopt dan met een foutmelding.

Sprites tijdens laden

Het komt wel eens voor dat u tijdens het draaien van een programma iets van tape of disk wilt laden. U heeft bijvoorbeeld een bestandenprogramma gemaakt waarmee u een bestand van tape wil laden. Of u wilt bijvoorbeeld een hiresplaatje van disk laden. Tot zover zijn er geen problemen.
Het laden gaat echter fout als er sprites op het beeld staan! U zult hiervan waarschijnlijk wel opkijken. Tijdens het laden moeten de sprites ook steeds op het beeld worden gezet door de computer. Het beeld wordt immers steeds opnieuw opgebouwd. Als er tijdens het laden sprites op het beeld staan, gaat er iets fout met interrupts. Het gevolg is dat het programma niet goed geladen wordt. Overigens hebben we het hier over laden van disk. Bij het laden van tape hebben we geen last van sprites op het beeld. Eenvoudigweg omdat tijdens het laden van tape het scherm wordt uitgezet. Bij laden van disk blijft het scherm echter aan en dan gaat het fout als er sprites op het beeld staan. De simpelste manier om dit probleem op te lossen, is de sprites uit zetten. Dit kan met POKE 53269,0.
U kunt ook zelf het beeld uitzetten. Dit kan gebeuren met bit 4 uit adres 53265. Als dit bit op 1 staat is het scherm aan. Met een 0 in dit bit kan het beeld worden uitgezet. POKE 53265,PEEK(53265) AND NOT 16 zet het beeld dus uit. Met POKE 53265,PEEK(53265) OR 16 kan het beeld dan weer aangezet worden.

Autostart

Anne Bezemer uit Papendrecht schreef ons een brief waarin hij vroeg of wij konden uitleggen hoe een programma zelfstartend kan worden gemaakt. Omdat er wellicht meer lezers geïnteresseerd zijn in dit onderwerp, zullen we in deze aflevering uitgebreid aandacht besteden aan het maken van deze programma's.
Een zelf-startend programma is een programma dat direct, nadat het is ingeladen, start. De gebruiker hoeft het programma dus niet te starten met het RUN commando. Dit heeft tot gevolg dat de gebruiker, na het laden van het programma, geen controle meer krijgt over de computer, waardoor het onmogelijk is om het programma te copieren door middel van het SAVE commando. In combinatie met een reset beveiliging, zoals die in de vorige aflevering van 'Tips & Trucs 64' is beschreven, is een zelf-startend programma redelijk beveiligd tegen copieren. 'Redelijk beveiligd' betekend dat Jan Modaal niet in staat is om het programma te kraken. Voor de doorgewinterde kraker is het kraken van deze beveiliging helaas een fluitje van twee cent.
We zullen nu een methode bespreken waarmee een programma zelf-startend kan worden gemaakt. We moeten daarbij onderscheid maken tussen machinetaal en Basic programma's. Eerst leggen we het zelf-startend maken van machinetaal programma's uit, omdat dit eenvoudiger is dan het zelf-startend maken van een Basic programma.
Om een programma zelf-startend te maken, moeten we uitzoeken wat de computer precies doet nadat het programma geladen is en voordat de gebruiker de controle over de computer terugkrijgt. Door daaraan iets te veranderen, kan ervoor gezorgd worden dat de computer het zojuist geladen programma runt in plaats van dat het de controle aan de gebruiker teruggeeft. Vlak voordat de gebruiker de controle over de computer terugkrijgt, wordt de melding 'READY.' op het scherm afgedrukt. Het afdrukken van deze melding gebeurt door indirect via adres 770 ($0302) te springen met de machinetaal instructie JMP ($0302). Deze instructie zorgt ervoor dat de computer het machinetaal programma dat op het adres, dat in de adressen 770 en 771 staat, uitvoert. Standaard staat in de adressen 770 en 771 het adres 42115. Omdat de adressen 770 en 771 RAM adressen zijn, kunnen we deze adressen veranderen. Als we er dus op de een of andere manier voor kunnen zorgen dat direct na het laden van het programma het startadres van dit programma in de adressen 770 en 771 staat, zal niet de routine die de READY melding afdrukt, worden gestart, maar zal het programma worden gestart. Dit kan met een paar Basic instructies gedaan worden.
Als eerste laadt u het machinetaal programma dat u zelf-startend wil maken in. Daarna rekent u het hi- en lo-byte van het SYS adres uit met de volgende formule:
HI=INT(adres/256):LO=adres-256*HI
Vervolgens tikt u de volgende Basic regel in:

PRINT "{SHIFT-CLR}":POKE 770,LO:POKE 771,HI:POKE 43,2:POKE 44,3:SAVE "programma",8,1

Na het saven van het programma krijgt u de controle over de computer niet meer terug, maar zal het programma starten. Het programma staat nu beveiligd op disk. Als u het inlaadt zal het programma zelf starten.
Wat doet de bovenstaande Basic regel nu precies? Eerst wordt het scherm gewist. Vervolgens wordt in de adressen 770 en 771 het startadres van het programma gepoked. Dan wordt in de adressen 43 en 44 het adres 770 gepoked. De adressen 43 en 44 geven aan wat het beginadres van een programma is; vanaf dit adres wordt gesaved bij het SAVE commando. Tenslotte wordt het geheugengebied, dat begint op adres 770 en eindigt bij het eindadres van het programma, gesaved. Op deze manier wordt dus de veranderde waarde van adressen 770 en 771 meegesaved. Als het programma nu wordt ingeladen, wordt deze waarde weer ingelezen op 770 en 771, zodat het programma na het laden start. Omdat vanaf adres 770 wordt gesaved, wordt ook het schermgeheugen (1024-2023) gesaved. Hierdoor zal alles dat tijdens het saven op het scherm staat, tijdens het laden van het programma weer zichtbaar worden. We wissen daarom

voor het saven het scherm, zodat de ingetikte POKE's tijdens het laden niet op het scherm komen te staan. Degene die het programma laadt, zou anders kunnen zien hoe het programma beveiligd is.
Om het programma op tape te saven moet de 8 bij het save commando in een 1 worden veranderd. U moet het programma dan wel in turbo saven. Zonder turbo werkt de beveiliging namelijk niet.
Een tweede manier om een machinetaal programma zelf-startend te maken is, door de adressen 806 en 807 ($0326 en $0327) te veranderen in plaats van 770 en 771. In deze adressen staat het startadres van de routine die een karakter op het scherm afdrukt. Deze beveiliging is beter dan de beveiliging via adres 770. Als het programma namelijk niet met het LOAD commando wordt geladen, maar met een machinetaal monitor, wordt er geen READY afgedrukt, waardoor het programma met de eerste beveiliging niet zou starten. De beveiliging via 806 werkt hierbij wel, omdat er wel karakters worden afgedrukt.
Om een machinetaal programma op deze manier zelf-startend te maken, kunt u weer gebruik maken van de Basic regel, die bij de eerste manier is gebruikt; u moet dan alleen de adressen 770 en 771 in respectievelijk 806 en 807 veranderen.
Als u gebruikt maakt van de tweede beveiliging, moet u er wel voor zorgen dat het programma, nadat het gestart is, de waarden in de adressen 806 en 807 weer hersteld, omdat het anders onmogelijk is om karakters af te drukken.
Om Basic programma's zelf-startend te maken moeten we nog iets meer doen, omdat een Basic programma niet met SYS adres, maar met RUN moet worden gestart. Voor het zelf-startend maken van Basic programma's hebben we de toetsenbordbuffer nodig. We gaan nu eerst een klein machinetaal programma maken dat de ASCII-codes voor R, U, N en {RETURN} in de toetsenbordbuffer zet. Nadat dit programma is uitgevoerd, wordt het commando RUN uitgevoerd. Als we dit machinetaal programma samen met het Basic programma saven dan is het probleem van het zelf-startend maken van een Basic programma gereduceerd tot het probleem van het zelf-startend maken van een machinetaal programma. Dit laatste probleem hebben we al opgelost. We moeten nu alleen nog kijken hoe we het machinetaal programma samen met het Basic programma kunnen saven. Als we het machinetaal programma op adres 900 zetten zal dit automatisch gebeuren, omdat we voor het zelf-startend maken van het machinetaal programma namelijk vanaf adres 770 of 806 moeten saven. Het machinetaal programa ziet er als volgt uit:

```
Labels Mnemonics Commentaar
       ldx #$03
lus    lda run,x    ;Zet 'run' in
       sta $0277,x  ;toetsenbordbuffer.
       dex
       bpl lus
       lda #$04     ;Zet waarde 4 in
       sta $c6      ;in adres 198.
       lda #$ca     ;Herstel adressen
       sta $0326    ;806 en 807.
       lda #$f1
       sta $0327
       rts
run    asc "run"
       .byt $0d
```

U kunt een Basic programma nu als volgt zelf-startend maken:
1. Laad het Basic programma in het geheugen.
2. Zet het machinetaal programma op adres 900.
3. Tik de volgende regel in:

PRINT "{SHIFT-CLR}":POKE 43,38:POKE 44,3:POKE 806,132:POKE 807,3:SAVE"naam",8,1

Testbeeld

Met behulp van een raster interrupt is het mogelijk om het beeldscherm in een aantal horizontale balken te verdelen. Zo kunnen de schitterendste demo's worden gemaakt. Elk van deze balken kan bijvoorbeeld een eigen achtergrond- en borderkleur worden gegeven; in de ene balk kan een prachtig hires plaatje worden afgebeeld; in een andere balk vliegen een aantal sprites rond en in weer een andere balk scrollt in een exotisch lettertype een lichtkrant, waarin wordt opgeschept over hoe mooi de demo wel is en hoe moeilijk het was om de demo te programmeren. In deze paragraaf geven wij een klein programma waarmee u de makers van dergelijke demo's kunt verbazen.
Dit programma verdeelt het scherm niet, zoals altijd, in een aantal horizontale balken maar in een aantal vertikale. Via de raster interrupt is dit niet mogelijk, omdat deze alleen maar werkt met horizontale beeldlijnen. Het programma werkt daarom zonder raster interrupt; het gebruikt zelfs geen andere interrupts. Helaas kunnen er met dit programma alleen maar een aantal kleurenbalken gemaakt worden, zonder enige tekst of sprites op het scherm; deze kleurenbalken lopen echter wel over het hele scherm, dus ook in de border, waardoor het scherm eruit ziet als een testbeeld. Wat het programma doet, klinkt nogal simpel, maar probeert u het maar eens te maken zonder verder te lezen. U zult dan zelf wel ontdekken dat het helemaal niet zo eenvoudig is, alhoewel?
Het programma werkt als volgt. Als het tekstscherm uitgeschakeld is, door in adres 53265 een 0 te poken (het hele scherm heeft dan de kleur van de border), en de interrupt is gestopt met het machinetaal commando SEI, dan duurt het opbouwen van een beeldlijn precies 63 cycles (een beeldlijn wordt puntje voor puntje van links naar rechts opgebouwd). Een cycle is een tijdseenheid die vaak in de informatica gebruikt wordt. Bij de Commodore 64 duren 1.000.000 cycles ongeveer een seconde. De machinetaal LDA #$getal instructie duurt 2 cycles. De STA $adres instructie duurt 4 cycles en de JMP $adres instructie duurt 3 cycles. In 63 cycles kunnen dus precies 10 keer een LDA en een STA en 1 keer een JMP instructie worden uitgevoerd. Na elke LDA en STA instructie is de computer weer een stukje verder met het opbouwen van een beeldlijn. Door nu met de LDA en de STA instructie 10 keer de kleur van de border (adres 53280) te veranderen, is de beeldlijn verdeeld in 10 verschillend gekleurde stukjes. Als daarna met de JMP instructie weer naar het begin van het programma wordt gesprongen, ontstaan er vertikale, gekleurde balken. Hieronder volgt de listing van het 'kinderachtig eenvoudige' programma:

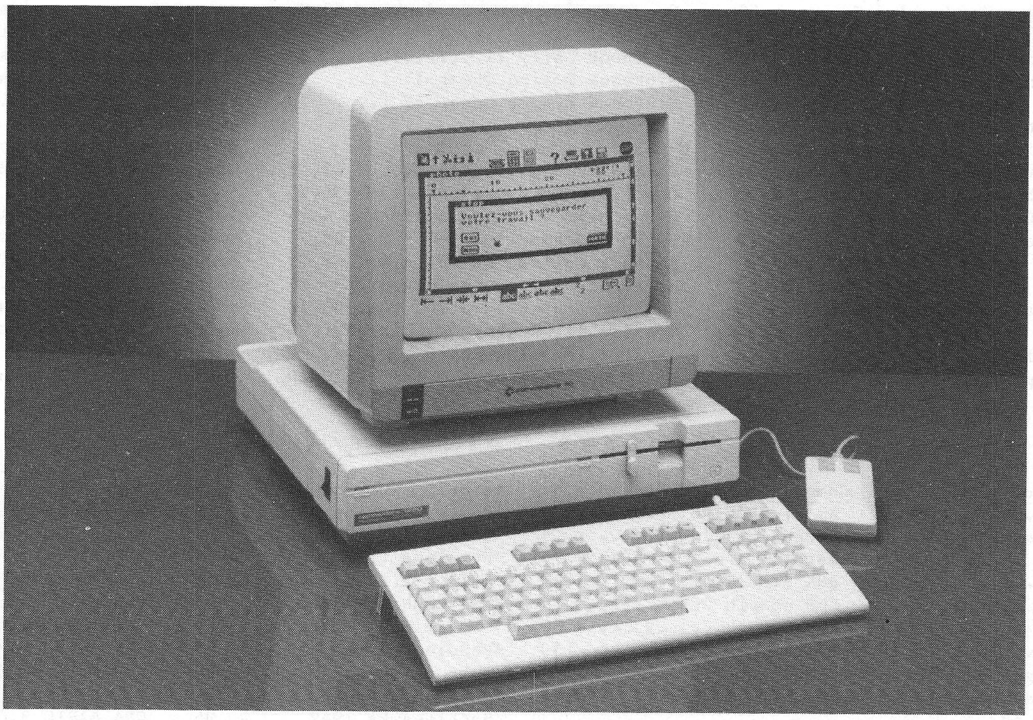
```
Labels Mnemonics Commentaar

    sei              ;Zet
    interrupt uit
    lda #$00
    sta $d011        ;Zet scherm
    uit
lus lda #$00         ;Verander de
    border
    sta $d020        ;10 keer van
    kleur.
    lda #$01         ;Omdat het
    hele
    sta $d020        ;scherm de
    kleur van
    lda #$02         ;de border
    heeft,
    sta $d020        ;veranderd het hele
    lda #$03         ;scherm van kleur.
    sta $d020
    lda #$04
    sta $d020
    lda #$05
    sta $d020
    lda #$06
    sta $d020
    lda #$07
    sta $d020
    lda #$08
    sta $d020
    lda #$09
    sta $d020
    mp lus           ;Start opnieuw.
```

Door op de RESTORE toets te drukken, nadat het programma gestart is, kunnen de balken verschoven worden.

Hylke Sprangers & Michel de Boer

Basic 8.



Eindelijk is er dan weer een Basic pakket dat de 80 cls mode ondersteund, En wat voor één !. Het pakket wordt geleverd op twee 5.25 diskettes. Iedere kant van de diskettes bevat een deel van het pakket en is AutoBootable. "Lees eerst de handleiding goed door, en ga daarna pas BASIC 8 installeren. Dit voorkomt veel ellende". Bent u ongeduldig, zoals wij waren, dan kan er wel eens wat fout gaan, want BASIC 8 moet eerst op vijf diskettes worden geïnstalleerd. Daarna kunt u de originelen veilig in de kast opbergen, en verder werken met de door het programma zelf aangemaakte kopie.

Wat is BASIC 8.

BASIC 8 is als het ware de uitgebreidere versie van de basic versie 7.0 die standaard in de C-128(D) zit. Het mooie van dit programma is dat het functies ondersteund in de 80 cls mode en dat houdt dus in dat u beschikt over een resolutie van 640 x 200 beeldpunten. Dus beschikt u niet over de mogelijkheid om in de 80 cls mode te werken dan heeft het geen nut om dit programma aan te schaffen, mits u van plan bent om een 80 cls monitor (RGB) aan te schaffen. En dan kunt u als nog beslissen het programma aan te schaffen, want de mogelijkheden zijn ongekend. Ok er zijn al diverse tools op de markt gekomen voor de 80 CLS ondersteuning, maar dit is toch veel beter, voor zover wij het hebben kunnen uitproberen.

HET OPSTARTEN VAN BASIC 8.0.

De bedoeling is dat als u het pakket hebt uitgepakt u als eerste de handleiding leest, de handleiding is vrij overzichtelijk maar wel in het Engels. Nadat de handleiding is doorgelezen hebt u al een idee van wat het programma allemaal kan. Nu doe je gewoon de masterflop in de drive en schakel uw computer aan, of reset de computer, en dan vervolgens de meldingen van het beeldscherm opvolgen.

Wanneer u het programma hebt opgestart dan is het mogelijk om een RUNTIME of een EDITOR DISK te creëren. Als u nu voor RUNTIME kiest dan is het mogelijk om een diskette te preperen om programma's te runnen. Het programma zal vragen om je 'Target' diskette in je drive te doen en daarna weer je Master diskette. Dit zal, als u geen snelladder hebt, enige minuten nemen, maar het is wel de moeite waard. Met de EDITOR DISK is het precies het zelfde als met de RUNTIME DISK alleen is deze diskette om je programma, zoals de naam al zegt, te editen, dus te creëren of aantepassen.

Het opstarten 2.

Alhoewel het programma copyrights heeft, is het programma niet uitgerust met een copieer beveiliging. Het programma komt zelfs met een ingebouwde backup mogelijkheid. De originele diskettes die u koopt zijn niet bedoeld om mee te werken. De eerste keer als u het programma opstart, moet u vijf lege diskettes bij de hand hebben. Dan is het de bedoeling om één van de originele BASIC 8 diskettes in de drive te plaatsen en vervolgens de computer in te schakelen of te resetten (hangt er van af of de computer al aan stond). Afhankelijk van welke Master Disk u in de drive heeft geplaatst verschijnt er een menu of een prompt. Disk 1 kant 1 heeft twee menu opties die u in staat stellen om een BASIC 8 Editor Work Disk of een BASIC 8 RunTime Disk te maken. U moet nu van elk diskette één maken. De andere drie kanten van de diskettes stellen u in staat om een BA-

SIC PAINT Work Disk, een BASIC 8 BasicWrite/BasicCalc Work Disk, of een Basic 8 Utility Work Disk te creeren. Als u nu alle vijf werk diskettes heeft gemaakt kunnen u de originele diskettes veilig in de kast worden opgebergen. Die kunt u een andere keer weer gebruiken om nog meer werk of run diskettes mee te maken. De BASIC 8 Editor Work Disk is voor het schrijven en veranderen van eigen BASIC 8 programma's. Deze disk mag niet worden weggegeven. Deze disk bevat het complete editor systeem. De BASIC PAINT Work Disk bevat een 80 kolloms kleuren tekenprogramma, en verder menu's, fonts en requestors. Het is een 'RunTime' programma, dat houd in dat het alleen de Runtime files bevat en u er dus niet mee kunt editen. Deze disk is een Autoboot disk dus alleen er in stoppen, computer resetten en klaar is kees. Het programma kan met muis of met joystick werken. Met de RunTime disk is het mogelijk om via de legale manier eigen geschreven of bijgeleverde software 'weg te geven' of 'te verkopen'. Deze bevat de Runtime files welke nodig zijn om BASIC 8 programma's te runnen (en ook deze is Autoboot). De BASIC 8 BasicWrite/Basiccalc Disk is ook een Runtime disk die diverse professionele programma's bevat die in BASIC 8 zijn geschreven. BasicWrite is een grafische tekstverwerker, die kan worden gebruikt om documenten te creëren met diverse fonts, zestien kleuren en zelfs plaatje in het document. BasicCalc is een kleine spreadsheet met de mogelijkheid om met de ingegeven data grafieken te maken. En de Basic 8 Utilities Disk is een verzameling van utilities, spelletjes en applicaties voor het maken van de meeste BASIC 8 software.

Configuratie.

Als vereiste bij BASIC 8 is dat u beschikt over (natuurlijk) de C-128(D), en op z'n minst één disk drive (1541, 1571, 1581 of één van hun 'broertjes'), en als laatste een RGB Monitor.
Eén van de grote voordelen is dat het programma de interne 64K RAM uitbreiding van de 8563 Video Display Chip (standaard ingebouwd in de C128D modellen, maar ook te krijgen voor de C128) ondersteund. Verder ondersteund het programma de externe RAM uitbreiding van de 1700 (128K RAM) en van de 1750 (512K RAM), de 1351 proportionele muis, de standaard joystick en de meeste Dot Matrix Printers.

Basic programma's.

De bijgeleverde basic programma's zijn volledig geschreven in BASIC 8 met toevoegingen van Basic 7.0. Bij het pakket zijn de volgende programma's bijgeleverd :
- BASIC PAINT
- BASIC WRITE
- BASIC CALC
- en WORKBENCH

Verder worden er nog diverse Basic games en Utilities geleverd.
Het totaal lijkt dan ook op 'The Works' (Amiga/PC).

Basic Paint.

Basic Paint is een 80 cls teken programma dat beschikt over zestien kleuren en een resolutie van 640 x 200 beeldpuntjes verder is het volledig in basic geschreven. Met dit programma is het mogelijk om tekeningen, brushes en iccons te maken, welke toepasbaar zijn in je eigen BASIC 8.0 programma's. Het programma biedt de standaard grafische functies (draw, circle, box en polygon). Verder is het mogelijk om tekst toe te voegen, gebieden te 'vullen', en de tekeningen uit te printen. Maar het pakket gaat nog verder zoals 3-D, zoom, cut en paste. De besturing gaat via de muis of joystick, waarbij je het pijltje uit acht verschillende vormen kunt kiezen. Het programma werkt in alle vijf verschillende grafische mode ondersteund door BASIC 8.0. U kunt kiezen uit monochroom of vier kleuren resolutie (8x16, 8x8, 8x4 en 8x2). Het programma ondersteund de 1700 en de 1750 externe RAM uitbreiding modules.

Basic Write.

Basic Write is een tekstverwerker in de sfeer van 'Desktop Publishing' die werkt met zestien kleuren en waarbij het mogelijk is om tekst te combineren met graphics. U kunt kiezen uit vijfentwintig verschillende tekst groten. U kunt ook gebruik maken van de standaard Commodore Grafische karakterset. Ondanks dat het een BASIC 8.0 programma is, is de snelheid niet mis. De tekeningen en brushes van BASIC PAINT kunnen hierin worden toegepast, dus indrukwekkende documenten. Verder heeft het programma alle functies die een goede tekstverwerker nodig heeft. En ook de nodige hoeveelheid printerdrivers is aanwezig.

Basic Calc.

BASIC CALC is een vrij eenvoudig te bedienen spreadsheet, en heeft genoeg in huis om er prettig mee te werken. Het werkveld is opgebouwd uit 780 cellen (26 rijen bij 30 kolommen). De rijen zijn genummerd van A tot Z, en de kolommen van 0 tot 29. De commando's worden ingevoerd op de commando lijn. En er is een mogelijkheid om de ingevoerde gegevens grafisch weer te geven.

Workbench.

Wat is dat die 'Workbench' ?. Nou dat zal ik uitleggen. Workbench is te vergelijken met een muis gestuurd menuprogramma. Dat houd dus in dat u vanuit dit programma diverse programma's die op de zelfde schijf staan op te starten. Het is ook mogelijk het programma te gebruiken met een joystick, maar dat word bij het opstarten keurig gevraagd. Het programma is te vergelijken met de eenvoudige versie van GEOS die ook geleverd is voor de C128(D). Het voordeel is natuurlijk dat dit programma gemaakt is in BASIC 8. Het programma gebruikt ook de reeds aanwezige Basic 7.0 Dus zeer gemakkelijk zelf aan te passen. Met de laatste versie is het mogelijk om printer drivers te installeren en is in staat om BASIC 8 tekeningen te laten zien op het beeld of uit te printen d.m.v. het pijltje te wijzen en dan aan te klikken. De Workbench is in staat om tot 4 disk drives te ondersteunen, inclusief de 1541, 1571, en 1581 drives. Bij het opstarten zal het programma zelf nagaan hoeveel drives er aanwezig zijn en zal zich daar op aanpassen. Het programma werkt met pull down windows en dat is vrij prettig om mee te werken.

Utilities.

De utilities die bij het programma worden geleverd laten de mogelijkheden van BASIC 8 zien en zoals al gezegd zijn die ongekend. Zo is er onderandere een programma dat 'Caleidoscoop' heet, het programma laat 128 kleuren op het beeld zien met een grafisch hoogstandje. Verder zijn er nog twee spelletjes op de schijf bijgevoegd namelijk Mastermind en Match. Ook is er een slideshow die alleen werkt met de 64K RAM Video Display Chip, maar zeker de moeite waard om te zien.

Conclusie.

Eén ding is zeker, de extra commando's t.o.v. Basic 7.0 zijn zeer goed te gebruiken. Want wie wil niet met een resolutie van 640 x 192 beeldpunten en 16 kleuren werken. De ondersteuning van de 80 cls mode levert dan ook grafische hoogstandjes op. De gebruiksaanwijzing laat op bepaalde punten het wel eens afweten. Het pakket BASIC 8 wordt geleverd door Free Spirit software inc. En als aanvulling op BASIC 8 is de BASIC 8 TOOLKIT te verkrijgen. Het is zeker de moeite waard om dit pakket tot U collectie toe te voegen.

JOHAN & JOHAN

Voorraad 128

```
Datahome Amsterdam

Datum : 20-11-89

-----------------------------------
Tabel leveranciers
-----------------------------------
001 .............................
002 .............................
003 .............................
004 .............................
005 .............................
006 .............................
007 .............................
008 .............................
009 .............................
010 .............................
011 .............................
012 .............................
013 .............................
014 .............................

 F1 = stoppen    F3 = Toevoeg
```

Nadat de Commodore 128 al een aantal jaren in gebruik is wordt de vraag naar software voor deze onderschatte computer steeds groter. Nu praten we niet over spelletjes, in de 64 mode zijn de spelen die er te gebruiken zijn niet te tellen maar over gebruiksprogramma's die alle mogelijkheden van de Commodore 128 volledig te benutten. Datahome is een leverancier van software die heeft ingezien dat deze goedkope computer zeer goed te gebruiken is voor het bedrijfsleven. Al eerder hebben we bekeken fakturering en kasboek, ditmaal zijn we met voorraadbeheer aan de slag gegaan.

Al bij het uitpakken valt op dat er erg veel zorg is besteedt aan deze software. Het programma bestaat uit een beveiligde diskette, en een prima handleiding. Deze handleiding is niet een gekopieerd Engelstalig blaadje, het zijn maar liefst 62 pagina's A5. En helemaal in het Nederlands geschreven. Een extra service die Datahome kan bieden is het aanpassen van de software aan uw eigen wensen. Voorraad-128 is een programma dat als opslag kan dienen voor de kontrole en het beheer van uw voorraden. Het unieke van dit pakket is het Elektronisch Kas Boekingen systeem, kortweg EKS. Dit is te vergelijken met een moderne kassa. Alleen hier behoeven geen prijzen te worden ingevoerd, er wordt gewerkt met artikelnummering. De voorraad wordt na elke handeling bijgewerkt en is dus altijd aktueel. Wordt er gevraagd of een bepaald artikel aanwezig is in het magazijn dan is het simpelweg intikken van de naam of de code voldoende. Niet alleen is er dan direkt te zien hoeveel er nog van dit artikel zijn, alle verdere gegevens zoals inkoopprijs, verkoopprijs, leverancier zijn op het scherm te lezen.

Wat is er nodig

tit pakket maakt gebruik van een Commodore 128-D en werkt alleen in de 80 koloms mode. Heeft U een 'gewone' C128 dan moet de diskdrive een 1571 zijn. Deze kan dubbelzijdig werken en dit is noodzakelijk voor de goede werking van het programma. Verder heeft U nog een printer nodig. Bijna alle printers die door de C 128 worden ondersteund kunnen worden gebruikt. De printer moet 136 karakters op een A4 pagina kunnen printen. Dit gebeurt in de zogenaamde condensed-mode. Bij de nieuwere printers gebeurt dit door software matig de code te versturen. Bij sommige oudere printers dient dit door het omzetten van één of meerdere schakelaartjes te gebeuren. Een kleuren monitor werkt prettig, maar is niet direkt noodzakelijk. Een monochrome monitor met 80 karakters werkt ook goed.

Het programma

Het programma wordt geleverd op een beveiligde diskette. Het is daarom voorzien van een uniek registratie nummer. Als U het programma krijgt ziet er een kaartje bij waarop een toegangscode is vermeld. Het programma wordt gestart door 'booten', op een andere manier laden kan schade toebrengen aan de diskette. Nu wordt er om de toegangscode gevraagd, U heeft driemaal de gelegenheid om de juiste code in te toetsen anders wordt het programma gereset. Deze programma schijf wordt niet gebruikt voor het wegschrijven van de gegevens. Hiervoor dient een dataschijf te worden aangemaakt. Heeft U de toegangscode juist ingetoetst dan wordt het programma verder geladen. Nu wordt er nog gevraagd om een paar belangrijke gegevens, de datum en de tijd. Deze zijn belangrijk om later eventuele mutaties op de juiste plaats te kunnen uitvoeren. Bij het eerste maal opstarten moet er eerst een dataschijf worden aangemaakt. Hierop kunnen maximaal 1500 voorraad gegevens worden bijgehouden. Heeft U er meer dan moet er een tweede of zelf meerdere dataschijven worden aangemaakt. Op deze schijf staan ook de persoonlijke gegevens van uw bedrijf. Buiten de naam en adres gegevens worden hier ook ingevuld de BTW percentages en de wijze van muteren. Een tweede handeling die we hier eenmalig, buiten het muteren, moeten uitvoeren is het maken van een lijst van leveranciers. Naam, adres en telefoon worden in de lijst verwerkt. Er kunnen 200 adressen van leveranciers worden gebruikt. De nummering gebeurt automatisch en het geheel is menu gestuurd. Nu zijn de standaard handelingen gebeurt en kan het programma worden gebruikt.
Het programma start met het hart, het centrale gedeelte van het programma, dit is het hoofdmenu. In dit menu heeft U acht verschillende keuze mogelijkheden. Algemene procedures, invoeren artikelen, wijzigen artikelen, printen overzichten, muteren transacties, voorraden bijwerken, opvragen artikelen en het programma afsluiten.

1. Algemene procedures.

In dit submenu zijn weer zes menu-onderdelen samen gebracht. De standaard gegevens hebben we hier reeds ingevoerd, maar kunnen via deze optie worden veranderd. Het financiële verslag geeft een overzicht van alle inkopen, verkopen, brutowinst maar ook de betaalde en ontvangen BTW. Een extra dataschijf wordt via de derde keuze gemaakt. Veel verstandiger is het om een backup te maken van de eerste dataschijf zodat leverancier gegevens niet nogmaals behoeven te worden ingevoerd. Een extra dataschijf, met andere gegevens kan hier natuurlijk ook worden aangemaakt. Reset databestanden brengt U terug naar de beginsituatie, dus alles wordt op 0 gezet. De leverancier gegevens kunnen hier worden bijgewerkt,

```
Copyright 1989                          Regnr:VR1003 v1.0
ROGRAMMA - VOORRAAD C128                Tijd.....: 10:10
---------------------------------------------------------
          ↑ = vorig blad    <- = vervolgblad
---------------------------------------------------------
.    ..............
.    ..............      Tabelgegevens
.    ..............
.    ..............
.    ..............
.    ..............      Maximale invoer - : 200
.    ..............      Reeds ingevoerd - : 000
.    ..............      Nog ruimte voor - : 200
.    ..............
.    ..............      ---------------------------
.    ..............
.    ..............
.    ..............
n    F5 = wijzigen   F7 = vervallen  Maak uw keuze
```

aangevuld of veranderd. Afsluiten van een periode gebeurt via de laatste optie uit dit menu. Dit afsluiten kan gebeuren per week, maand of per kwartaal alle opgeslagen gegevens worden hier getotaliseerd.

2. Invoeren artikelen.

In dit submenu zijn ook weer zes menuonderdelen ondergebracht. Via deze optie worden alle artikelen ingevoerd. Er kunnen hier zoals reeds eerder gezegd 1500 artikelen worden gebruikt. De andere mogelijkheden zijn wijzigen, opvragen op nummer, op alfabet en ook op nummer. Via deze optie worden ook de eventuele adres stickers uit geprint.

3. Wijzigen artikel.

Dit onderdeel is bijna het zelfde als menu optie twee, U kunt hier alle ingevulde gegevens op een eenvoudig manier aanpassen.

4. Printen overzichten.

In dit menu zijn zeven onderdelen verzameld. Zij hebben allemaal betrekking op het maken van overzichten. Voordat U gaat printen krijgt U eerst een overzicht van instellingen van de meest voorkomende printers. Ook wordt hier gekontroleerd of de printer wel aanstaat, zodat er geen foutmelding kan komen en het programma zou vastlopen. Bij het uitprinten van een artikelen lijst wordt er eerst gesorteerd. Dit kan zowel op artikelnaam als op nummer geschieden. Het sorteren gaat redelijk snel, 1000 records in ongeveer 2 minuten. Een mutatielijst is een lijst met overzichten met veranderingen die reeds verwerkt zijn. Ook in dit onderdeel kunt U werken met de EKS. Printen van leveranciers en het printen van een verkooplijst spreken voor zich. Ook via dit menu kunt U etiketten uitprinten, dit kan zowel met als zonder de naam van de leveranciers.

5. Muteren transacties.

Dit menu onderdeel is de kern van het EKS systeem. Veranderingen worden via dit menu ingebracht. Het geheel bestaat uit vijf submenu's. Invoeren, wijzigen, opvragen, kasboek bijwerken en erg belangrijk mutaties verwerken. Procedure 5, verwerken gegevens, is de eigenlijke dag afsluiting. Alle (winkel) verkopen worden naar het mutatie bestand weggeschreven. In dit menu kunt U weer een volledig overzicht krijgen van de stand van zaken, en dit op elk gewenst moment. In het EKS systeem kunnen 200 mutaties verwerkt worden.

6. Voorraden bijwerken.

Dit menu onderdeel wordt er gekozen als er artikelen bijkomen. Dit is bijvoorbeeld het geval als er artikelen worden ingekocht. Ook kunnen er voorraden uit het magazijn worden gehaald zonder dat er is verkocht. Op het scherm wordt weergegeven o.a. weergegeven wat de oude- en de nieuwe voorraad is.

7. Opvragen artikelen.

Dit onderdeel wordt gebruikt als U gegevens over een artikel wilt hebben. Dit is bijvoorbeeld het geval als er een artikel moet worden ingekocht, en U weet niet meer precies wie de leverancier van dat bewuste artikel is.

8. Programma beeindigen.

Voor dat het programma daadwerkelijk wordt beeindigd moet altijd eerst de dag worden afgesloten. Dit is niet nodig als er geen mutaties zijn geweest via het EKS. De tweede optie, een backup maken is wel gewenst. Gebeurt er iets met uw kostbare dataschijf, b.v. uw frisdrank gaat er over heen, dan is het prettig te weten dat er nog een kopie van uw schijf is. Hierdoor voorkomt U dat er weken werk verloren gaat. Hierbij is wel het programma DOS-Shell noodzakelijk dat U bij de aanschaf van uw computer heeft gekregen.

Conclusie:

Voorraad 128 is een programma dat voor een klein of middelgroot bedrijf een prima alternatief kan zijn voor een dure computer configuratie. Het programma werkt goed, de prijs- kwaliteits vehouding is gunstig, redelijk snel, maar wat belangrijker is het is erg betrouwbaar. Dit alles gekombineerd met het feit dat Datahome een gerenommeerd software leverancier is, dus de garantie en begeleiding is aanwezig, maakt dat dit pakket een goede keus is.

Prijs Voorraad-128 ƒ 299,50
Leverancier Datahome Amsterdam
Inlichtingen 020-837367

RBG

Amiga C cursus

```
 }
 return(start);
}

UBYTE *allocmw(size,type)
 LONG size,type;
{
 extern struct memwatch *mainwatch;

 struct memwatch *newmw;
 UBYTE *watch=AllocMem(size,type);

 if(mainwatch==NULL)
 {
  if(!(newmw=mainwatch=(struct memwatch *)
   AllocMem(sizeof(struct
    memwatch),MEMF_CLEAR|MEMF_PUBLIC)))
  {
   printf("help");
   Exit(21);
  }
 }
 else
 {
  newmw=endmw(mainwatch);
  if(!(newmw=newmw->nextmem=(struct memwatch *)
   AllocMem(sizeof(struct
    memwatch),MEMF_CLEAR|MEMF_PUBLIC)))
```

```
  {
   printf("help");
   Exit(22);
  }
 }
 newmw->size=size;
 newmw->pointer=watch;
 return(watch);
}

VOID freemw(pointer,size)
 UBYTE *pointer;
 LONG size;
{
 extern struct memwatch *mainwatch;

 struct memwatch *seeker,*lastseek=NULL;

 if(mainwatch!=NULL)
 {
```

Na vele maanden absentie zijn we weer terug. Speciaal nu voor de weer donkerder wordende winterdagen is hier het vervolg en tevens een nieuwe start van onze Amiga C Kursus.

Het heeft enige tijd geduurt voordat we weer terug waren. De bedoeling was dat deze aflevering al eerder afgedrukt zou worden, echter een technisch mankementje verhinderde dit. Kwestie van een te los vingertje die te gemakkelijk JA antwoordt op 'verwijder CKURS.TXT (j/n)'. Goed, laten we het niet meer over deze fout hebben, let's play ball!
In de vorige afleveringen, 1, 2 en 3, hebben we ons beperkt tot standaard C. Hierin komt nu verandering. Geheel in overeenstemming met de kop bij dit artikel, beginnen we nu met programmeren van de Amiga in C.

Intuition

Waarmee kunnen we beter beginnen dan met Intuition te beschrijven. Intuition is dat deel van het besturingssysteem waarmee de meeste gebruikers van de Amiga mee in aanraking komen. Het zorgt voor het op de juiste manier omzetten van diverse soorten invoer, zoals muis, joystick, diskdrive, toetsenbord, etcetera. Deze gegevens worden voorbereid en op een bepaalde manier worden deze dan ter beschikking gesteld. Een tweede aspect van Intuition is de programmering van de beeldschermuitvoer door middel van Windows, lees vensters, en Screens, lees schermen. Samenvattend kunnen we dus zeggen dat Intuition, op een gebruikersvriendelijke manier, een medium is tussen de Amiga en de gebruiker, dat bent u!

We kunnen een grove onderverdeling maken van het grafische besturingssysteem. Dit vormt tevens de rode draad door deze C cursus heen. Al eerste en de meest aan het oppervlakte liggende laag, Intuition. Als de functies behorende bij Intuition zijn te vinden in de 'intuition.library'. Intuition, op zijn beurt, is op een lager gelegen laag gebaseerd, de graphics-primitives. De routines die deze vormen kunt u vinden in de 'graphics.library'. Tenslotte zijn we dan aangeland op het machineniveau. Op dit niveau zult u de machine rechtstreeks aan moeten spreken.
In de 'intuition.library' vinden we diverse subroutines die op Intuition niveau werken. Hoe kunnen we deze aanspreken? Zoals u mischien al vermoed, de 'intuition.library' moet eerst geopend worden. Hiervoor gebruiken we een functie die uit de 'exec.library' komt. Nu zult u zich afvragen, deze 'exec.library' moet toch ook geopend worden? Waar en niet waar. Waar in zoverre dat deze ook geopend dient te worden en niet waar daar deze automatisch bij het opstarten van het sys-

teem al geopend wordt. Machinetaal programmeurs zullen dit weten doordat zij de pointer naar deze library op locatie $000004 in het geheugen kunnen vinden. Voor ons is dit van minder belang. We kunnen de routines van de 'exec.library' gewoon oproepen zonder een library te openen. Andere libraries, daarentegen, dienen wel geopend te worden. Hiervoor levert 'exec.library' de functie OpenLibrary(). Voorzien van enige parameters zal deze functie de pointer naar de in OpenLibrary() opgegeven library teruggeven. Wat voor parameters moeten worden meegegeven?
LibraryNaam=OpenLibrary(LIBRARY NAAM,VERSIE);
Eerst maar een voorbeeld. Stel u wilt de 'intuition.library' openen. Het volgende voorbeeld doet dit.

IntuitionBase=(struct IntuitionBase *)
OpenLibrary("intuition.library",0L);

In de variabele IntuitionBase die van het type 'struct IntuitionBase' is zal nu de pointer staan naar de 'intuition.library'. De declaratie van de variabele IntuitionBase luidt dan ook als volgt;

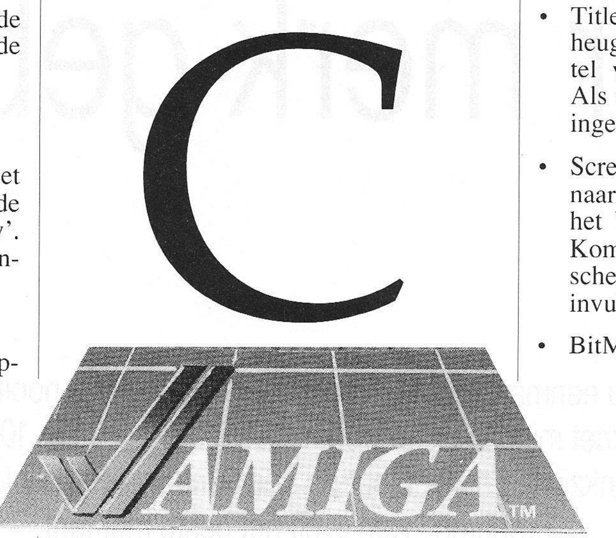
struct IntuitionBase *IntuitionBase;

We willen de pointer naar de library opslaan dus dienen we de variabele ook als de pointer NAAR het type te declareren. Lijkt even een lastige woordenbrij, is echter zeer simpel na wat experimenteren.
Als we éénmaal de 'intuition.library' geopend hebben staat ons een heel scala van functies en procedures tot onze beschikking. Even een kleine opsomming van wat er te vinden is.

OpenScreen(), CloseScreen(),
OpenWindow(), CloseWindow(),
DisplayAlert(), DrawBorder(),
AddGadget(), MoveScreen(),
MoveWindow() en nog vele andere.

De NewWindow structuur

Hebben we éénmaal de 'intuition.library' geopend dan ligt de halve Amiga wereld voor ons open. Toch beginnen we bij het begin. Vanuit BASIC was het openen van een Window of Screen zeer simpel. Roep het commando op en geef enkele parameters mee en het hele verhaal wordt wel op je beeldscherm gedumpt. Nee, helaas in C is het allemaal niet zo simpel.
Een Window of venster beschikt over twee structuren. De eerste is de NewWindow structuur. Deze dient u te gebruiken, voordat u een Window geopend heeft, als u een Window WILT openen. De tweede structuur is de Window structuur. Heeft u eenmaal een Window geopend en wilt u toegang tot de gegevens krijgen van dit Window dan zult u een Window structuur moeten gebruiken. Hierover straks meer. Eerst nu weer terug naar de NewWindow structuur. U kunt deze niet zomaar bovenaan in het programma declareren. De structure wordt opgebouwd in het include bestand 'intuition/intuition.h'. Include deze met het #include commando en u kunt alle de in dit bestand beschreven structures en macro's gebruiken. Om een idee te krijgen hoe de NewWindow structure er uit ziet, bekijk figuur 1 eerst eens. Zoals u ziet dienen er nogal enige elementen ingevuld te worden. We gaan de elementen in logische volgorde behandelen.



- LeftEdge en TopEdge, moeten de coordinaten van resp. de linker- en bovenhoek bevatten. De x-waarden zullen afhankelijk van de gekozen schermresolutie zijn, doch zullen variëeren van 0 tot maximaal 600. Ook de y-waarden zijn afhankelijk van de schermresolutie en liggen in het bereik van 0 tot maximaal 500.
- Width en Height, hierin geeft u op resp. hoe breed en hoog het venster moet worden. Let erop, de waarden van Width+LeftEdge mag niet groter zijn dan het maximale aantal pixels waarover u in de huidige schermmode de beschikking heeft. Ditzelfde geld ook voor Height+TopEdge.
- DetailPen en BlockPen, ze dienen de kleurcode van de voorgrond- en achtergrondkleur te bevatten.
- IDCMPFlags, wilt u gebruik maken van de IDCMP faciliteit van de Amiga dan dient u hierin op te geven welke mogelijkheden u wilt gebruiken. Hiervoor zijn bepaalde FLAGS gecreërd die u in kunt vullen. Hierover meer in een volgende aflevering.
- Flags, ook voor het Window zijn aparte mogelijkheden gereserveerd, zoals de mogelijkheid tot windowvergroting,-versleping, -sluiting, etcetera. Voor meer informatie omtrent deze flags zie het aparte kader bij dit artikel.
- FirstGadget, kan een pointer naar het eerste door u zelf gedefinieerde gadget bevatten.
- CheckMark, kan een pointer naar een Image structuur bevatten. In deze Image structuur wordt een nieuw activerings haakje gedefinieerd die u aan kunt treffen in menu's.
- Title, dient een pointer naar een geheugendeel te bevatten waarin de titel van het Window is opgeslagen. Als u geen titel wilt moet hier NULL ingevuld worden.
- Screen. Hierin dient u een pointer naar het scherm te plaatsen waarin het Window geplaatst moet worden. Komt het Window in het WorkBench scherm terecht dan mag u hier NULL invullen.
- BitMap, dient een pointer naar een eigen gedefinieerde BitMap structuur te bevatten. Hiermee reserveert u op eigen wijze geheugen voor het Window. MinWidth en MinHeight, bepalen de minimale breedte en hoogte van het Window als u gebruik maakt van de WindowSizing optie.
- MaxWidth en MaxHeight, idem als vorige alleen nu bepaalt u de maximale breedte en hoogte.

Nu we alle elementen globaal besproken hebben wordt het tijd voor een klein voorbeeld. Bekijk programma 1 eerst eens. Op zich een heel simpel programmaatje om een Window te openen. Wat voor functie wordt nu gebruikt om een Window te openen? Afgezien van de functie OpenLibrary() is het er maar ééntje. De functie OpenWindow() opent een Window waarvan u de eigenschappen gedefinieerd heeft in een NewWindow structuur. Het zal u wel duidelijk zijn, de functie dient dus te weten welke NewWindow strucuur hij moet gebruiken. De enigste parameter die OpenWindow() heeft is dus een pointer naar een NewWindow structuur. Kort gezegd,

pointer naar een Window =
OpenWindow(pointer naar een NewWindow);

Zoals u uit bovenstaande definitie kunt zien, geeft de functie een pointer terug. Deze pointer slaat u op in een variabele van het type 'struct Window'. Begint het al een beetje te dagen? Deze Window

structuur wordt ingevuld door de OpenWindow() functie. Wilt u nu toegang hebben tot een geopend Window dan dient u een pointer te hebben naar de Window structuur die bij het Window hoort. Nog even kort samengevat, een NewWindow structuur vult u met de eigenschappen van een Window zoals u het wilt hebben. Vervolgens geeft u deze NewWindow structuur door aan de functie OpenWindow(). Een Window structuur gebruikt u alleen als u eigenschappen wilt weten van een al geopend Window. Bekijk programma 1 nu nog een keer vanuit dit oogpunt.

De Window structuur

We hebben al uitgelegd waarvoor u een Window structuur zult of wilt gebruiken. Hoe ziet zo'n Window structuur er uit? Figuur 2 geeft een opsomming van de elementen van een Window structuur. In tegenstelling tot de NewWindow structuur gaan we niet de elementen van de Window structuur bespreken. Wilt u hier meerdere details over weten, bekijk dan de literatuurlijst aan het einde van deze aflevering. Hier zult u meer informatie over boeken kunnen vinden.

De Window structuur is eigenlijk nog belangrijker dan de NewWindow structuur daar u de Window structuur in alle operaties, die Windows manipuleren, nodig zult hebben. De structuur wordt opgebouwd in het include bestand 'intuition/intuition.h'. Vergeet dus niet deze eerst te includen. Gemakkelijker is het include bestand 'intuition/intuitionbase.h' te includen daar deze toch al nodig is voor de IntuitionBase declaratie. In het bestand 'intuition/intuitionbase.h' wordt het include bestand 'intuition/intuition.h' geincluded. Nadat u de bestanden geincluded heeft, komen we aan bij de globale variabelen declaratie. Bij eenvoudige programma's kunt u 'struct Window' en 'struct NewWindow' wel globaal declareren. Welke declaraties zijn voor het openen van een Window persé noodzakelijk? Ten eerste natuurlijk IntuitionBase, daar u anders geen toegang zou kunnen krijgen tot de 'intuition.library'. struct IntuitionBase *IntuitionBase; zorgt ervoor dat deze variabele bekend is. Verder hebben we een Window structuur nodig dus volgt er een soortgelijke declaratie, struct Window *MijnWindow;. Nu komen we bij de declaratie van NewWindow aan. U mag deze laten auto-initialiseren, dwz. u vult de waarden meteen in bij het declareren van de NewWindow structuur. Een klein voorbeeld.

```
struct NewWindow NW = {
  0,0,640,256,
  0x00,0x01,
  VANILLAKEY,
    SMART_REFRESH|BORDERLESS|RM
    BTRAP,
  NULL,NULL,NULL,NULL,NULL,
  0,0,0,0,CUSTOMSCREEN };
```

Hierdoor worden alle elementen in logische volgorde ingevuld. Er is echter wel een nadeel aan. U dient hierbij strikt de volgorde van de structure aan te houden.

De flags behorende bij NewWindow.

Bij elk venster behoren een aantal flags. Met deze flags is het mogelijk één of meerdere opties te selecteren die Intuition biedt. De opties zijn grofweg onder te verdelen in een aantal groepen. Flags die betrekking hebben op de Gadgets. Flags die bepalen hoe met de data van het scherm wordt omgesprongen. Vervolgens hebben we een groep die de flags omvat die bepalen wat voor soort venster het wordt en als laatste de overige flags die niet bij de voorgaande groepen in te delen zijn.

Gadget flags

° WINDOWDRAG: Door op te titelbalk te 'klikken' kunt u het venster verschuiven.

° WINDOWDEPTH: Hierdoor komen twee gadgets tevoorschijn waarmee u het venster afwisselend op de voor- en achtergrond kunt plaatsen.

° WINDOWCLOSE: Als u een venster weg wilt kunnen klikken gebruikt u deze optie. Het zorgt ervoor dat het afsluitgadget opgenomen wordt. Het venster zal dan nog niet meteen verdwijnen als u hierop klikt. Het zal een melding aan het programma doorgeven waarop het programma het venster af moet sluiten.

° WINDOWSIZING: Door deze optie verschijnen de 'sizing' gadgets onderin het venster waarmee het venster in grootte te variëren is.

° SIZEBRIGHT: Het 'sizing' gadget wordt in de rechterrand afgebeeld.

° SIZEBBOTTOM: Het 'sizing' gadget wordt in de onderrand van het venster geplaats.

Refresh flags

° SIMPLE_REFRESH: Intuition doet in dit geval helemaal niets. Het programma krijgt een signaal waarna het, het venster moet herstellen.

° SMART_REFRESH: Zodra een venster geheel of gedeeltelijk wordt bedekt door een ander venster, zal het programma de grafische gegevens van het bedekte deel opslaan en weer tevoorschijn halen als het bedekkende venster weer verdwijnt.

° NOCAREREFRESH: In de twee voorgaande modi zal het programma gewaarschuwd worden als het venster hersteld (moet) word(t)/(en). Deze melding onderdrukt u met NOCAREREFRESH.

Venstermodi flags

° BACKDROP: Het betreffende venster zal altijd het achterste venster zijn.

° SUPER_BITMAP: Het venster is veel groter dan op het scherm zichtbaar is.

° BORDERLESS: Intuition tekent geen rand (=border) om het venster.

° GIMMEZEROZERO: Het 'binnenblad' van het venster en de border worden gescheiden behandeld. Tekenen door de border behoord hierdoor tot de mogelijkheden. Een voordeel is dat deze methode een snellere vensterafhandeling oplevert.

Overige flags

° ACTIVATE: In deze modus zal het venster meteen actief worden. Invoer zal dus ook meteen via dit venster geschieden.

° REPORTMOUSE: De actuele positie van de muis zal voortdurend doorgegeven worden.

° RMBTRAP: Right Mouse Button TRAP. Het signaal van de rechter muisknop zal gelijk zijn aan het signaal van de linker muisknop.

Wilt u geen gebruik maken van de vensterflags vul dan 'NULL' in.

U mag dus niet de elementen door elkaar heen halen daar anders toekenningen in de verkeerde elementen terecht komen. Een andere oplossing zou dus zijn de elementen pas in de main() of andere procedure cq. functie te initialiseren. Op deze manier vult u de elementen op dezelfde wijze in zoals u anders een structure invult. Ook hiervan een kort voorbeeld.

```
struct NewWindow NW;
main()
{
 NW.LeftEdge=0;
 NW.Flags=WINDOWDRAG|ACTIVATE;
 NW.TopEdge=0;

....
```

Zoals u ziet mag u hierbij de elementen wel door elkaar heen halen doordat u gedwongen bent op te geven welk element u in wilt vullen.

Programma 3 geeft een voorbeeld hoe men het aantal Windows kan tellen door gebruik te maken van het element NextWindow uit de Window structuur. Net zo lang tot dit element een NULL bevat, zal de routine door de gelinkte lijst van Window structuren springen. Aan het einde geeft het, het aantal gevonden Windows terug.

Window sluiten

Als u éénmaal een Window geopend heeft zal u onvermijdelijk ook weer een Window af willen sluiten. Het sluiten van een Window is nog gemakkelijker dan het openen ervan. U weet u nog te herinneren dat u een pointer naar een Window structuur terugkreeg bij het openen van een Window. Deze pointer heeft u opgeslagen in een variabele van het type 'struct Window'. Deze pointer heeft u nu ook weer nodig bij het sluiten van het Window. Om een Window te sluiten dient u gebruik te maken van de functie CloseWindow(). Geef in CloseWindow() als parameter de pointer naar het Window mee op en het Window zal afgesloten worden. Een voorbeeld. Stel u heeft een Window geopend waarvan u de teruggekregen pointer opgeslagen heeft in MijnWindow die u gedeclareerd heeft als 'struct Window *MijnWindow;'. Sluit het Window nu weer door op te geven;
CloseWindow(MijnWindow);
Dit is alles, meer is er niet aan!
Het Window zal gesloten worden en u bent klaar. Probeer nu geen functies meer los te laten op het gesloten window want het systeem zal zeker 'crashen'.

Figuur 1. De NewWindow structuur.

```
struct NewWindow = {
    SHORT LeftEdge;
    SHORT TopEdge;
    SHORT Width;
    SHORT Height;
    UBYTE DetailPen;
    UBYTE BlockPen;
    ULONG IDCMPFlags;
    ULONG Flags;
    struct Gadget
   *FirstGadget;
    struct Image *CheckMark;
    UBYTE *Title;
    struct Screen *Screen;
    struct BitMap *BitMap;
    SHORT MinWidth;
    SHORT MinHeight;
    SHORT MaxWidth;
    SHORT MaxHeight;
    USHORT Type;
  };
```

Windows en hun functies

Zoals u nu gezien heeft kunt u een aantal mogelijkheden van Windows in de NewWindow structuur instellen. Toch zal het gebeuren dat u de Windows, nadat ze geopend zijn, wilt aanpassen. Dit is mogelijk doordat er een aantal functies ontworpen zijn die specifieke uitwerkingen hebben op Windows. We gaan ze in alfabetische volgorde behandelen.

ActivateWindow(pointer naar Window): als u een Window wilt activeren, gebruik dan deze functie.

ClearPointer(pointer naar Window): deze functie hersteld de muispointer weer in z'n originele, dwz, door de system-configuration bepaalde, vorm.

CloseWindow(pointer naar Window): sluit het opgegeven Window.

CloseWorkBench(): een wat vreemde eend in de bijt. Toch wel handig, vooral als u maar over 512K beschikt. Hiermee sluit u de WorkBench. Let op, er mag hierna ook geen uitvoer meer naar het standaarduitvoer scherm van de WorkBench gaan.

MoveWindow(pointer naar Window, deltaX, deltaY): hiermee verschuift u een Window over deltaX en deltaY pixels. Let op, het Window mag niet buiten de grenzen van het scherm vallen.

OpenWindow(pointer naar NewWindow): opent een Window met in NewWindow vermelde eigenschappen.

OpenWorkBench(): hiermee opent u de WorkBench weer.

SetPointer(pointer naar Window, pointer naar Image structure (voor pijl), hoogte van sprite, breedte van sprite, x-aanwijspunt, y-aanwijspunt): als u de muispointer in het Window van vorm wilt laten veranderen, dan kunt u gebruik maken van deze functie.

SetWindowTitles(pointer naar Window, pointer naar tekstbuffer voor Window, pointer naar tekstbuffer voor Screen): als u, na opening van het Window, de titel nog wilt veranderen dan is dit mogelijk door gebruik te maken van deze functie. Ook de schermtitel is hiermee te veranderen.

SizeWindow(pointer naar Window, deltaX, deltaY): deze functie is het equivalent van de rechteronderhoek gadget. Hiermee kunt u immers de grootte van het Window instellen. Geef in deltaX en deltaY op, het aantallen pixels dat u het Window kleiner wilt hebben. Let op, het Window mag niet kleiner worden dan 0 en niet groter worden dan wat het huidige scherm toelaat.

WindowLimits(pointer naar Window, MinWidth, MinHeight, MaxWidth, MaxHeight): met deze functie kunt u flexibel

Figuur 2. De Window structuur.

```
struct Window = {
     struct Window *NextWindow;
     SHORT LeftEdge;
     SHORT TopEdge;
     SHORT Width;
     SHORT Height;
     SHORT MouseX;
     SHORT MouseY;
     SHORT MinWidth;
     SHORT MinHeight;
     USHORT MaxWidth;
     USHORT MaxHeight;
     ULONG Flags;
     struct Menu *MenuStrip;
     UBYTE *Title;
     struct Requester
    *FirstRequest;
     struct Requester
    *DMRequest;
     SHORT ReqCount;
     struct Screen *WScreen;
     struct RastPort *RPort;
     BYTE BorderLeft;
     BYTE BorderTop;
     BYTE BorderRight;
     BYTE BorderBottom;
     struct RastPort
    *BorderRPort;
     struct Gadget
    *FirstGadget;
     struct Window *Parent;
     struct Window *Descendant;
     BYTE PtrHeight;
     BYTE PtrWidth;
     BYTE XOffset;
     BYTE YOffset;
     ULONG IDCMPFlags;
     struct MsgPort *UserPort;
     struct MsgPort
    *WindowPort;
     struct IntuiMessage
    *MessageKey;
     UBYTE DetailPen;
     UBYTE BlockPen;
     struct Image *CheckMark;
     UBYTE *ScreenTitle;
     SHORT GZZMouseX;
     SHORT GZZMouseY;
     SHORT GZZWidth;
     SHORT GZZHeight;
     UBYTE *ExtData;
     BYTE *UserData;
     struct Layer *WLayer;
     struct TextFont *IFont;
  };
```

Instructies bij window_demo1.c

Programmanaam	:	window_demo1.c
Programmeertaal	:	C
Gebruikt programma	:	Aztek C v3.6a
Opties	:	cc window_demo1.c +L ln window_demo1.o -lc32
Aanroep	:	window_demo1

Programma 1. Demonstratie van een simpel venster

```
#include "functions.h"
#include "exec/types.h"
#include "intuition/intuitionbase.h"

struct IntuitionBase *IntuitionBase;
struct Window *TestWindow;
struct NewWindow NW = {
     10,10,200,100,0x00,0x01,NULL,
     WINDOWDRAG|WINDOWDEPTH|WINDOWSIZING,
     NULL,NULL,(UBYTE *)"C_Info testwindow",
     NULL,NULL,10,10,640,256,WBENCHSCREEN
    };

main()
{
 register int i;

 if(!(IntuitionBase=(struct IntuitionBase *)
   OpenLibrary("intuition.library",0L)))
     Quit(5,"Intuition is niet te openen!\n");
 if(!(TestWindow=(struct Window *)
   OpenWindow(&NW)))
     Quit(5,"Window niet te openen!\n");

 for(i=0;i<1000000;i++);

 Quit(0,"Programma goed verlopen.\n");
}

Quit(hoe,waarom)
int hoe;
char *waarom;
{
 printf("%s",waarom);

 if(TestWindow!=NULL)
   CloseWindow(TestWindow);
 if(IntuitionBase!=NULL)
   CloseLibrary(IntuitionBase);

 exit(hoe);
}
```

Instructies bij window_demo2.c

Programmanaam	:	window_demo2.c
Programmeertaal	:	C
Gebruikt programma	:	Aztek C v3.6a
Opties	:	cc window_demo2.c +L ln window_demo2.o -lc32
Aanroep	:	window_demo2

Programma 2. Windows en hun functies

```
#include "functions.h"
#include "exec/types.h"
#include "intuition/intuitionbase.h"

struct IntuitionBase *IntuitionBase;
struct Window *TestWindow;
struct NewWindow NW = {
        10,10,300,130,0x00,0x01,

    NULL,WINDOWDRAG|WINDOWSIZING|WINDOWDEPTH,
         NULL,NULL,(UBYTE *)"Commodore INFO",
         NULL,NULL,10,10,640,200,
         WBENCHSCREEN };

main()
{
 register int i;

 if(!(IntuitionBase=(struct IntuitionBase *)
   OpenLibrary("intuition.library",0L)))
     Quit(5,"Geen intuition.library!\n");

 if(!(TestWindow=(struct Window *)
   OpenWindow(&NW)))
     Quit(5,"Dit window wil niet openen!\n");

 for(i=0;i<1000000;i++);

 SetWindowTitles(TestWindow,"Even een andere
   titel");
 for(i=0;i<1000000;i++)
   {
    MoveWindow(TestWindow,(SHORT)1,(SHORT)1);

    SizeWindow(TestWindow,(SHORT)-1,(SHORT)-1);
   }

 for(i=0;i<1000000;i++);

 Quit(0,"Programma goed verlopen.\n");
}

Quit(hoe,waarom)
int hoe;
char *waarom;
{
 printf("%s",waarom);

 if(TestWindow!=NULL)
    CloseWindow(TestWindow);
 if(IntuitionBase!=NULL)
    CloseLibrary(IntuitionBase);

 exit(hoe);
}
```

de Window grenzen voor WindowSizing gebruik aanpassen. Ze vult opnieuw de waarden van MinWidth, MinHeight, MaxWidth en MaxHeight in.
WindowToBack(pointer naar Window): met WindowToBack() voert u dezelfde functie uit als wanneer u op het rechtergadget in de rechterbovenhoek van het scherm 'klikt'. Het Window 'valt' hierdoor naar achteren.
WindowToFront(pointer naar Window): idem als hierboven, alleen nu wordt het Window voor de anderen geplaatst.

Bij het gebruik van deze functies dient u erom te denken dat de Windows niet buiten de grenzen van het scherm komen. Doet u dit wel, dan zal het systeem tevoorschijn komen met een 'Guru'!
Nu we u de specifieke Window functies hebben laten zien, wordt het tijd voor een concreet voorbeeld. Typ programma 2 in. Compile en link het en voer het uit. Leuk effect dachten we zo. Bekijk nu het programma eens. Zoals u ziet wordt in dit programma bescheiden gebruik gemaakt van enkele Window functies.

Als laatste opmerking

We willen u nog op één punt wijzen. Denk erom dat u de 'intuition.library' pas sluit als u het Window al gesloten heeft. Wilt u het Window sluiten NADAT u 'intuition.library' gesloten heeft dan zal het resultaat de wellicht overbekende en te vaak geziene 'Guru-meditatie' zijn.
Voor deze keer rest ons niets meer te zeggen daar wij uitgesproken zijn. Wat kunt u in de volgende aflevering verwachtten. Screens. Schermen dus. Hoe opent u ze. Wat voor bewerkingen voert u er mee uit, etcetera. Dit alles natuurlijk nog steeds op Intuitionniveau. Goed, dat was het, tabee, ciao, buenos nog wattes of gewoon tot de volgende keer!

Literatuurlijst

Berry, J.T. (1986), Inside the Amiga with C, Howard W.Sams & Comp., Indiana (USA) ISBN 0-672-22468-2
Bleek, Jennrich & Schultz (1988), Het grote Amiga C boek, Bruna uitg., Utrecht ISBN 90-229-3483-7
Jennrich, Trapp & Weltner (1987), Het supergrafiekboek voor de Amiga, Bruna uitg., Utrecht ISBN 90-229-3470-5

JOHAN & JOHAN

Instructies bij window_demo3.c

Programmanaam	:	window_demo3.c
Programmeertaal	:	C
Gebruikt programma	:	Aztek C v3.6a
Opties	:	cc window_demo3.c +L ln window_demo3.o -lc32
Aanroep	:	window_demo3

Programma 3. Een voorbeeld met al bestaande Windows

```
#include "functions.h"
#include "exec/types.h"
#include "intuition/intuitionbase.h"
struct IntuitionBase *IntuitionBase;
ScFWindows(ParWindow)
struct Window *ParWindow;
{
/* Zolang de pointer naar het volgende venster, te vinden in  */
/* ParWindow-NextWindow, ongelijk aan 0 (NULL) is moet door  */
/* worden gegaan met tellen. Dit wordt bereikt door met behulp*/
/* van recursie opnieuw, met doorgave van de pointer naar het */
/* volgende venster, de routine in te springen. Bij iedere lus*/
/* wordt de terug te geven waarde steeds met 1 verhoogt.      */

 if(ParWindow-NextWindow==NULL)
   return(0);
 else
   return(1+ScFWindows(ParWindow-NextWindow));
}

main()
{
 SHORT NumWindows;

/* Open de library intuition.library                          */

 if(!(IntuitionBase=(struct IntuitionBase *)
   OpenLibrary("intuition.library",0L)))
     Quit(5,"Geen intuition.library\n");

/* Door gebruik te maken van de verwijzing in struct          */
/* IntuitionBase wordt het kinderspel het aantal windows te   */
/* tellen. Maak gebruik van IntuitionBase-ActiveScreen.       */
/* In de structure Screen wordt een verwijzing gegeven naar   */
/* het eerste venster. De volledige verwijzing luidt dan:     */
/* IntuitionBase-ActiveScreen-FirstWindow                     */

 NumWindows=ScFWindows(IntuitionBase-ActiveScreen-FirstWindow);
 printf("Aantal windows = %2d\n",NumWindows+1);
/* En verlaat vervolgens met behulp van de functie Quit()     */
/* het programma weer.                                        */

 Quit(0,"Programma succesvol verlopen\n");
}
Quit(hoe,waarom)
int hoe;
char *waarom;
{
/* Een standaard foutafhandelings procedure. Ook bij het      */
/* verlaten van het programma wordt de routine aangeroepen. In*/
/* programma's van deze grootte geen absolute noodzaak echter */
/* het maakt het programma veel beter leesbaar. Bij grote     */
/* programma's en veel declaraties is het een absolute 'must' */
/* aangezien men anders 'verzuipt' in het programma en ontoe- */
/* laatbaar veel geheugen gebruikt wordt.                     */

 printf("%s",waarom);

 if(IntuitionBase!=NULL)
   CloseLibrary(IntuitionBase);
 exit(hoe);
}
```

VIDIO Beelden 75

Checksum C-64

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.
2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtikken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...'komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven.

Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.

Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)

De laatste tijd wordt er weer veel gebeld zodat U nogwel eens in gesprek krijgt, daarom houdt uw vraag kort, vermeld in welk blad het desbetreffende artikel stond. Heeft U veel vragen, of zis uw vraag erg uitgebreid, doe het dan schriftelijk, zodat we veel mensen op de maandag avond te woord kunnen staan.

```
1   rem ******************************
2   rem basic loader "SYNTAX.CHECKSUM"
3   rem na de commando's "run" en "new"
4   rem blijft dit programma in het ge-
5   rem heugen. laad het te testen pro-
6   rem gramma en tik daarna sys 49152.
7   rem ******************************
10  i=49152  :rem beginadres
20  reada:ifa<0then40:rem data ingelezen
30  pokei,a:i=i+1:b=b+a:goto20
40  if b<>16844thenprint"[SHIFT-CLR]fo
    ut[SPACE]in[SPACE]dataregels!":b=0
    :end
50  poke49184,148:poke49185,192
55  i=49300
60  read a: ifa<0then80
70  pokei,a:b=a+b:i=i+1:goto60
80  if b<>20068thenprint"[SHIFT-CLR]fo
    ut[SPACE]in[SPACE]dataregels![SPAC
    E](vanaf[SPACE]regel[SPACE]240)":b
    =0:end
90  print"data[SPACE]is[SPACE]weggezet"
95  print"checksum[SPACE]testen[SPACE]
    met[SPACE]sys49152"
100 data 165,43,166,44,133,163,134,164
    ,169,147
110 data 32,210,255,160,0,240,3,32,73,192
120 data32,73,192,208,1,96,32,225,255,208
130 data 3,76,116,164,32,81,192,32,73,192
140 data 240,12,201,32,240,247,24,101,
    167,133
150 data 167,76,37,192,166,167,169,0,1
    32,168
160 data 32,205,189,169,13,32,210,255,
    164,168
170 data 76,17,192,200,208,2,230,164,1
    77,163
180 data96,162,0,189,123,192,240,6,32,210
190 data 255,232,208,245,32,73,192,170
    ,32,73
200 data 192,132,168,32,205,189,162,3,
    169,32
210 data 32,210,255,202,208,250,169,0,
    133,167
220 data 164,168,96,82,69,71,69,76,32,0
230 data  -1
240 data 165,197,201,3,240,7,201,4,240
250 data 6,76,148,192,76,34,192,169
260 data 147,32,210,255,76,161,192
270 data -1
```

** EINDE LISTING checksum 64 **

REGEL	1	249	REGEL	100	183
REGEL	2	84	REGEL	110	158
REGEL	3	105	REGEL	120	232
REGEL	4	2	REGEL	130	183
REGEL	5	246	REGEL	140	96
REGEL	6	152	REGEL	150	96
REGEL	7	249	REGEL	160	127
REGEL	10	157	REGEL	170	71
REGEL	20	64	REGEL	180	223
REGEL	30	38	REGEL	190	73
REGEL	40	57	REGEL	200	79
REGEL	50	14	REGEL	210	109
REGEL	55	251	REGEL	220	106
REGEL	60	192	REGEL	230	225
REGEL	70	42	REGEL	240	16
REGEL	80	244	REGEL	250	163
REGEL	90	245	REGEL	260	92
REGEL	95	237	REGEL	270	22

Print out C-64 met o.a. Spreadsheet

Lettertekenaar 64

LET OP, dit programma werkt niet met een cardridge !! Ronald Deen, de maker van dit programma schrijft dat hij deze utilitie heeft ontwikkeld omdat hij bij het schrijven van programma's vaak karakters moest veranderen. Het programma is zo geschreven dat het alleen met een diskdrive werkt.

```
0     poke53280,0:poke53281,0:input"[SHI
      FT CLR]instructies";a$:ifa$="j"ora
      $="ja"then126
1     poke53280,0:poke53281,0:a=0:b=0:c=
      0:d=0:e=0:f=0:g=0:h=0:print"[SHIFT
       CLR]":goto77
2     poke52,56:poke56,56:clr
3     poke56334,peek(56334)and254
4     poke1,peek(1)and251
5     fori=375to511:pokei+14336,peek(i+5
      3248):nexti
6     poke1,peek(1)or4
7     clr:poke56334,peek(56334)or1
8     gosub33:poke808,234
9     print"[SHIFT-CLR]":da=0:poke53272,
      31:poke808,234
10    print"[SHIFT-CLR]":forq=dato255:po
      ke53280,1:poke53281,1
11    print"[CTRL-1][HOME][7xCRSR-UP]+++
      +++++1[6xSPACE][CTRL-9]YZ[SHIFT-+]
      [COM--][SHIFT--][COM-PI][COM-*][SH
      IFT SPACE][CTRL-0][17xSPACE]";
12    print"++++++++2[31xSPACE]";
13    print"++++++++3[SPACE][CTRL-9]HIJK
      LM[CRSR-RIGHT]N[CRSR-RIGHT]ORSTUVW
      [CRSR-RIGHT]X[CTRL-0][12xSPACE]";
14    print"++++++++4[10xSPACE][CTRL-9]P
      QQQQQQQ[CTRL-0][12xSPACE]";
15    print"++++++++5[31xSPACE]";
16    print"++++++++6[31xSPACE]";
17    print"++++++++7[31xSPACE]";
18    print"++++++++8[31xSPACE]";
19    print"12345678[32xSPACE]"
20    gosub105
21    goto89
22    poke14336+8*q+0,a(0):poke56096+q,2
23    poke14336+8*q+1,a(1):gosub81
24    poke14336+8*q+2,a(2)
25    poke14336+8*q+3,a(3)
26    poke14336+8*q+4,a(4)
27    poke14336+8*q+5,a(5)
28    poke14336+8*q+6,a(6)
29    poke14336+8*q+7,a(7):poke1824+q,q
30    print"[HOME][10xCRSR-DOWN][CTRL-1]
      ":goto79
31    printq"_";a(0);",";a(1);",";a(2);"
      ,";a(3);",";a(4);",";a(5);",";a(6)
      ;",";
32    printa(7);:nextq
33    readcn:ifcn<0thenreturn
34    fori=0to7:readcd
35    poke14336+8*cn+i,cd:nexti
36    goto33
37    data200,31,48,103,104,104,103,48,3
      1
38    data201,224,48,152,24,24,152,48,22
      4
39    data202,3,7,15,3,3,3,3,31
40    data203,63,99,195,195,99,63,3,255
41    data204,60,102,195,102,60,102,195,
      126
42    data205,255,195,134,12,126,48,96,1
      92
43    data206,0,0,0,128,148,204,164,196
44    data207,102,195,195,102,102,60,24,
      24
45    data208,24,24,15,7,0,0,0,0
46    data209,0,0,255,255,0,0,0,0
47    data210,51,99,99,99,51,51,31,14
48    data211,60,24,25,49,51,99,97,240
49    data212,120,204,6,6,6,12,140,248
50    data213,62,51,99,102,124,192,192,1
      92
51    data214,60,22,24,49,49,97,99,243
52    data215,120,236,198,158,182,102,19
      8,6
53    data216,3,15,30,30,60,48,128,192
54    data217,0,0,0,142,136,140,136,238
55    data218,0,0,0,238,68,68,68,68
56    data219,0,0,0,236,138,204,138,234
57    data220,0,0,0,14,4,100,4,4
58    data221,0,0,0,234,138,204,138,234
59    data222,0,0,0,234,142,206,142,234
60    data223,0,0,0,68,170,238,170,170
61    data224,0,0,0,198,166,198,160,166
62    data44,0,0,0,0,0,3,3,6
63    data32,0,0,0,0,0,0,0,0
64    poke14336+8*43 ,255
65    poke14336+8*43 +1,129
66    poke14336+8*43 +2,129
67    data31,17,49,255,255,49,17,0,0
68    data-1
69    poke14336+8*43 ,255
70    poke14336+8*43 +1,129
71    poke14336+8*43 +2,129
72    poke14336+8*43 +3,129
73    poke14336+8*43 +4,129
74    poke14336+8*43 +5,129
75    poke14336+8*43 +6,129
76    poke14336+8*43 +7,255:return
77    gosub64:print"[CTRL-2][SHIFT-CLR][
      11xCRSR-DOWN][9xCRSR-RIGHT]made[SP
      ACE]by[SPACE]yuiopia[SPACE]"
78    print"[HOME]":goto2
79    print"[40xSPACE]";:gosub64
80    print"[2xCRSR-UP]":goto31
81    poke14336+8*43 ,0
82    poke14336+8*43 +1,0
83    poke14336+8*43 +2,0
84    poke14336+8*43 +3,0
85    poke14336+8*43 +4,0
86    poke14336+8*43 +5,0
87    poke14336+8*43 +6,0
88    poke14336+8*43 +7,0:fori=0to500:ne
      xti:return
89    forkl=0to8:forpy=1024+(kl*40)to103
      1+(kl*40)
90    if peek(py)=32thenyp=py-(1023+(kl*
      40)):goto92
91    goto93
92    onypgoto94,95,96,97,98,99,100,101
93    nextpy:a(kl)=qu:qu=0:nextkl:goto22
94    qu=128:goto93
95    qu=qu+64:goto93
96    qu=qu+32:goto93
```

```
97  qu=qu+16:goto93
98  qu=qu+8:goto93
99  qu=qu+4:goto93
100 qu=qu+2:goto93
101 qu=qu+1:goto93
102 poke53272,21:input"[SHIFT-CLR]naam";ji$:ji$=ji$+",seq,write":gh=q*8
103 open2,8,2,ji$:forzz=14336to14336+gh:print#2,peek(zz):nextzz:poke53272,31
104 close2:da=gh/8:goto10
105 v=53248:pokev+21,4:poke2042,13:pokev+41,14
106 poke832,60:poke835,60:poke838,255:poke841,255:poke844,255:poke847,60
107 poke850,60
108 xx=24:yy=50
109 jk=peek(56320)
110 ifjk=119thenxx=xx+8:ifxx>88thenxx=88
111 ifjk=123thenxx=xx-8:ifxx<24thenxx=24
112 ifjk=126thenyy=yy-8:ifyy<50thenyy=50
113 ifjk=125thenyy=yy+8:ifyy>114thenyy=114
114 ifjk=111thenpoke1024+((xx-24)/8)+(((yy-50)/8)*40),32
115 ifjk=109thenpoke1024+((xx-24)/8)+(((yy-50)/8)*40),43
116 geta$:ifa$=chr$(13)thenreturn
117 ifa$="s"thenprint"[SHIFT-CLR]":run7
118 ifa$="w"then102
119 ifa$="r"thengoto122
120 ifa$="p"thengoto125
121 pokev+4,xx:pokev+5,yy:goto109
122 poke53272,21:input"[SHIFT-CLR]welk[SPACE]bestand?";b$:b$=b$+",seq,read"
123 open3,8,8,b$:fori=0to2046:input#3,g:poke14336+i,g:ifst=0thennexti
124 close3:poke53272,31:da=i/8-1:goto10
125 input"[CTRL-1]";ol$:goto121
126 print"[SHIFT-CLR][CTRL-2]letter-tekenaar[SPACE]is[SPACE]een[SPACE]programma[SPACE]waarmeeje[SPACE]je[SPACE]eigen[SPACE]karakter[SPACE]set[SPACE]kan
127 print"maken[SPACE]door-middel[SPACE]van[SPACE]de[SPACE]joystick[SPACE]in[SPACE]port2[SPACE]te[SPACE]doen[SPACE]en[SPACE]op[SPACE]de[SPACE]knop[SPACE]";
128 print"te[SPACE]drukken[SPACE]haal[SPACE]je[SPACE]een[4xSPACE]vlakje[SPACE]weg.als[SPACE]je[SPACE]knop[SPACE]en[SPACE]naar[SPACE]achteren[SPACE]";
129 print"tegelijk[SPACE]doet[SPACE]komt[SPACE]er[SPACE]weer[SPACE]een[SPACE]vlakje[3xSPACE]bij,zodat[SPACE]je[SPACE]fouten[SPACE]";
130 print"verbeteren[SPACE]kan[SPACE]![4xSPACE]als[SPACE]je[SPACE]<return>[SPACE]drukt[SPACE]worden[SPACE]alle[SPACE]lege
131 print"vlakjes[SPACE]1[SPACE]stip[SPACE]en[SPACE]die[SPACE]stippen[SPACE]tesamen[3xSPACE]vormen[SPACE]het[SPACE]tekentje[SPACE]dat[SPACE]u";
132 print"[SPACE]had[SPACE]gemaakt[SPACE]8[SPACE]maal[SPACE]zo[SPACE]klein.[SPACE]het[SPACE]cijfer[SPACE]voor[SPACE]de[SPACE]pijl[2xSPACE]is[SPACE]de[SPACE]poke-";
133 print"waarde[SPACE]van[SPACE]het[SPACE]teken.[SPACE]als[SPACE]u[2xSPACE]<p>[SPACE]indrukt[SPACE]kunt[SPACE]u[SPACE]bijvoorbeeld[SPACE]de[SPACE]";
134 print"a[SPACE]of[SPACE]de[SPACE]b[SPACE]indrukken[SPACE]om[SPACE]te[SPACE]kijken[SPACE]wat[SPACE]voor[4xSPACE]teken[SPACE]u[SPACE]daarvan[SPACE]heeft";
135 print"[SPACE]gemaakt.je[SPACE]kan[SPACE]ookalle[SPACE]waarden[SPACE]naar[SPACE]disk[SPACE]saven[SPACE]en[SPACE]laden,[2xSPACE]zodat";
136 print"[SPACE]niet[SPACE]alle[SPACE]waarden[SPACE]verloren[SPACE]gaan[3xSPACE]als[SPACE]je[SPACE]de[SPACE]computer[SPACE]uitzet";
137 print"[2xCRSR-DOWN][7xCRSR-RIGHT]<return>"
138 geta$:ifa$=""then138
139 print"[SHIFT-CLR]"
140 print"druk[SPACE]<return>[SPACE]als[SPACE]je[SPACE]klaar[SPACE]bent[SPACE]met[SPACE]de[16xSPACE]letter"
141 print"[5xCRSR-RIGHT]<s>......als[SPACE]je[SPACE]overnieuw[SPACE]wilt[19xSPACE]beginnen"
142 print"[5xCRSR-RIGHT]<p>......als[SPACE]je[SPACE]de[SPACE]tekens[SPACE]wil[20xSPACE]uitproberen"
143 print"[5xCRSR-RIGHT]<w>......als[SPACE]je[SPACE]alle[SPACE]waarden[SPACE]wil[17xSPACE]wegschrijven"
144 print"[5xCRSR-RIGHT]<r>......als[SPACE]je[SPACE]waarden[SPACE]wilt[21xSPACE]inlezen"
145 print"[5xCRSR-RIGHT][2xCRSR-DOWN]doe[SPACE]de[SPACE]joystick[SPACE]in[SPACE]port2[SPACE]!!"
146 print"[5xCRSR-DOWN][7xCRSR-RIGHT]<return>"
147 geta$:ifa$=""then147
148 goto 1
```

EINDE LISTING letter-tekenaar

Checksum letterteken 64

regel 0	248
regel 1	4
regel 2	62
regel 3	42
regel 4	127
regel 5	88
regel 6	28
regel 7	151
regel 8	41
regel 9	171
regel 10	84
regel 11	164
regel 12	162
regel 13	205
regel 14	160
regel 15	165
regel 16	166
regel 17	167
regel 18	168
regel 19	129
regel 20	35
regel 21	250
regel 22	115
regel 23	113
regel 24	67
regel 25	69
regel 26	71
regel 27	73
regel 28	75
regel 29	101
regel 30	93
regel 31	117
regel 32	170
regel 33	134
regel 34	207
regel 35	98
regel 36	239
regel 37	103
regel 38	122
regel 39	119
regel 40	59
regel 41	213
regel 42	192
regel 43	20
regel 44	165
regel 45	166
regel 46	214
regel 47	199
regel 48	245
regel 49	217
regel 50	183
regel 51	255
regel 52	239
regel 53	228
regel 54	10
regel 55	99
regel 56	17
regel 57	153
regel 58	8
regel 59	1
regel 60	221
regel 61	32
regel 62	215
regel 63	200
regel 64	85
regel 65	48
regel 66	49
regel 67	137
regel 68	225
regel 69	85
regel 70	48
regel 71	49
regel 72	50
regel 73	51
regel 74	52
regel 75	53
regel 76	254
regel 77	56
regel 78	229
regel 79	73
regel 80	38
regel 81	233
regel 82	196
regel 83	197
regel 84	198
regel 85	199
regel 86	200
regel 87	201
regel 88	182
regel 89	96
regel 90	155
regel 91	245
regel 92	183
regel 93	34
regel 94	34
regel 95	65
regel 96	60
regel 97	62
regel 98	15
regel 99	11
regel 100	9
regel 101	8
regel 102	146
regel 103	65
regel 104	219
regel 105	146
regel 106	122
regel 107	198
regel 108	203
regel 109	90
regel 110	23
regel 111	1
regel 112	10
regel 113	104
regel 114	65
regel 115	74
regel 116	147
regel 117	75
regel 118	119
regel 119	253
regel 120	254
regel 121	232
regel 122	214
regel 123	222
regel 124	214
regel 125	170
regel 126	88
regel 127	50
regel 128	182
regel 129	16
regel 130	5
regel 131	110
regel 132	143
regel 133	238
regel 134	188
regel 135	33
regel 136	56
regel 137	36
regel 138	105
regel 139	112
regel 140	175
regel 141	8
regel 142	73
regel 143	93
regel 144	20
regel 145	177
regel 146	87
regel 147	105
regel 148	184

Aanvallen 64

Aanvallen is op zich een eenvoudig spel. Het is de bedoeling dat U door middel van het op het juiste moment indrukken van de spatiebalk de 'viques' door de dijk laat gaan. Maar er zijn slechts 25 kansen om dit te doen. Na het runnen van dit spel drukt U op de spatiebalk om het spel te starten. Dit kan zowel met als zonder instructies. Het spel is gemaakt door Lennart van Walsum uit Hove. (dit ligt in Belgie)

```
1    rem ******************
2    rem **  aanvallen    **
3    rem **  ---------    **
4    rem **     door       **
5    rem **                **
6    rem **  lennart vw   **
7    rem **  en hugo c.   **
8    rem **                **
9    rem ******************
10   poke53280,0:poke53281,0:poke53272,22
20   print"[SHIFT-CLR][COM-1][CTRL-9]AANVALLEN":gosub390
30   print"[2xCRSR-DOWN][COM-3]bij[SPACE]dit[SPACE]spelletje[SPACE]is[SPACE]er[SPACE]de[SPACE]bedoeling[SPACE]dat"
40   print"[COM-7]U,[SPACE]doormiddel[SPACE]van[SPACE]een[SPACE]druk[SPACE]op[SPACE]de[SPACE]space-"
50   print"[CRSR-DOWN][COM-2]balk,[SPACE]de[SPACE]'VIQUES'[SPACE]door[SPACE]de[SPACE]dijk[SPACE]laat[SPACE]gaan"
60   print"[CTRL-8][CTRL-6]maar[SPACE]pas[SPACE]op,[SPACE]U[SPACE]hebt[SPACE]slechts[SPACE]25[SPACE]kansen[SPACE]om"
70   print"[COM-4]uw[SPACE]inzicht[SPACE]en[SPACE]aanval[SPACE]te[SPACE]beproeven[SPACE]!!!!!"
80   print"[CRSR-DOWN][CTRL-3]wilt[SPACE]U[SPACE]spelen[SPACE][CTRL-9]m[CTRL 0]et[SPACE]of[SPACE][CTRL-9]z[CTRL-0]onder[SPACE]instrukties[COM 2]":g=25
90   poke198,0:wait198,1:gett$
100  ift$="m"thenprint"[HOME]":goto130
110  ift$="z"thenprint"[SHIFT-CLR]":goto130
120  ift$<>""then90
130  fort=2023-159to2023:poket,104:nextt
140  fort=1064+rnd(0)*40to1503:poket,122
150  geta$:ifa$<>""thengoto200
160  poket,32:nextt
170  fort=1503to1064step-1:poket,122
180  geta$:ifa$<>""then200
190  poket,32:nextt:goto140
200  g=g-1
210  print"[HOME][COM-7]Je[SPACE]mag[SPACE]nog[SPACE]"g"[CRSR-LEFT][2xSPACE]zetten[SPACE]doen..[COM-3]"
220  ifg=1thenprint"[HOME][COM-7]Je[SPA
```

```
     CE]mag[SPACE]nog[SPACE]"g"[SPACE]z
     et[SPACE]doen...[COM-3]":goto250
230  ifg<>-1then250
240  ifg=-1then310
250  ifpeek(t+40)=104thenpoket,32:poket
     +40,32:goto140
260  ifpeek(t+40)=104thenpoket,32:poket
     +40,32:goto140
270  poket,32:t=t+40:ift<2024thenpoket,
     122:goto250
280  poke53272,21
290  print"[SHIFT-CLR][CTRL-9]mooi[SPAC
     E]zo,[SPACE]goed[SPACE]gedaan[SPAC
     E]![SPACE]wil[SPACE]je[SPACE]nog[S
     PACE]eens[SPACE]?[CTRL-0][CRSR-DOW
     N][17xSPACE][CTRL-9](j/n)"
300  goto350
310  poke53272,21:print"[SHIFT-CLR][5xS
     PACE][CTRL-9]zwak,[SPACE]zwak,[SPA
     CE]zwak,[SPACE]zwak[SPACE]!!!!!!![
     CTRL-0]":
320  gosub390
330  print"[2xCRSR-DOWN]nog[SPACE]een[S
     PACE]speee":gosub410:print"[CRSR-U
     P][11xCRSR-RIGHT]l[SPACE]":gosub41
     0
340  print"[CRSR-UP][12xCRSR-RIGHT]letj
     e[SPACE]??":gosub410
350  poke198,0:wait198,1:geta$
360  ifa$="j"thenrun
370  ifa$="n"thenend
380  ifa$<>""then350
390  fort=1to1000:nextt:poke198,0:wait1
     98,1
400  return
410  fort=1to500:nextt
420  return
430  end
```

EINDE LISTING aanvallen 64

Checksum aanvallen 64

regel 1	131	regel 210	108
regel 2	201	regel 220	204
regel 3	204	regel 230	80
regel 4	107	regel 240	155
regel 5	55	regel 250	134
regel 6	248	regel 260	134
regel 7	110	regel 270	95
regel 8	55	regel 280	41
regel 9	131	regel 290	1
regel 10	137	regel 300	33
regel 20	120	regel 310	16
regel 30	17	regel 320	41
regel 40	15	regel 330	24
regel 50	149	regel 340	24
regel 60	58	regel 350	160
regel 70	99	regel 360	97
regel 80	15	regel 370	91
regel 90	179	regel 380	215
regel 100	52	regel 390	199
regel 110	193	regel 400	142
regel 120	187	regel 410	1
regel 130	249	regel 420	142
regel 140	155	regel 430	128
regel 150	154		
regel 160	140		
regel 170	42		
regel 180	17		
regel 190	228		
regel 200	28		

Bowlen 64

Bowlen is een databestand wat speciaal is toegespitst om alle gegevens van een bowling competitie bij te houden. Zelfs een gemmiddelde en de hcp kan worden berekent. De maker van dit programma is de Heer Lageman uit Amsterdam. Het programma is volledig menugestuurd, dus een ieder moet er mee kunnen werken.

```
1    rem bowlen door s.g. lageman.
2    rem
3    rem       amsterdam.
4    rem
5    rem hcp berekend over de laatste 21
     games, indien hcp over alle games
6    rem dan de regels 101 t/m 106 laten
     vervallen.
7    rem  verdeling gegevens speler  x,
     y,z        x = speler     y = week
8    rem  s(10,52,8) 10spelers 52 weken
9    rem  s(x,0,0) computer nummer speler
10   rem  s(x,0,1) aantal keren input
11   rem  s(x,0,2) hoogste serie
12   rem s(x,0,3) hoogste game
13   rem z0=datum          z1=game 1
14   rem z2=game 2         z3=game 3
15   rem z4=game 4         z5=totaal
16   rem z6=sub. totaal z7=gemiddelde
17   rem z8=handicap
18   rem
19   rem set up
20   rem
21   dims(10,52,8),n$(11):n$(0)="[CTRL-
     9][SPACE]menu[2xSPACE][CTRL-0]":v=
     53280:pokev,6:pokev+1,6
22   h$="":k$=",":f1$="[CTRL-9][COM-1][
     SPACE]f1[SPACE][CTRL-0][CTRL-2]":n
     $(11)="alle[SPACE]spelers"
23   f2$="[CTRL-9][COM-1][SPACE]f2[SPAC
     E][CTRL-0][CTRL-2]":f3$="[CTRL-9][
     COM-1][SPACE]f3[SPACE][CTRL-0][CTR
     L 2]":f4$="[CTRL-9][COM-1][SPACE]f
     4[SPACE][CTRL-0][CTRL-2]":f5$="[CT
     RL 9][COM-1][SPACE]f5[SPACE][CTRL-
     0][CTRL-2]"
24   f6$="[CTRL-9][COM-1][SPACE]f6[SPAC
     E][CTRL-0][CTRL-2]":f7$="[CTRL-9][
     COM-1][SPACE]f7[SPACE][CTRL-0][CTR
     L 2]":f8$="[CTRL-9][COM-1][SPACE]f
     8[SPACE][CTRL-0][CTRL-2]"
25   p$="[COM-1]-----------------------
     ----------------[CTRL-2]"
26   rem
27   rem hoofd menu
28   rem
29   print"[SHIFT-CLR][HOME]"
30   printtab(6)"[COM-1]UDDDDDDDDDDDDDD
     DDDDDDDDDDDDDI"
31   printtab(6)"G[CTRL-2][SPACE]**[SPA
     CE]bowlen[SPACE]bestand[SPACE]disk
     [SPACE]**[COM-1][SPACE]H"
32   printtab(6)"JFFFFFFFFFFFFFFFFFFFFF
     FFFFFFK"
33   printtab(14)"[CTRL-9][SPACE]menu[S
     PACE][CTRL-0][2xCRSR-DOWN]"
34   printtab(7)f1$"[SPACE]team[SPACE]/
```

```
    [SPACE]score[SPACE]invoer[CRSR-DOW
    N]"
35  printtab(7)f2$"[SPACE]oude[SPACE]g
    egevens[SPACE]ophalen[CRSR-DOWN]"
36  printtab(7)f3$"[SPACE]hcp[SPACE]en
    [SPACE]score[SPACE]overzicht[CRSR-
    DOWN]"
37  printtab(7)f4$"[SPACE]score[SPACE]
    overzicht[SPACE]printen[CRSR-DOWN]"
38  printtab(7)f5$"[SPACE]gegevens[SPA
    CE]wijzigen[CRSR-DOWN]"
39  printtab(7)f6$"[SPACE]gegevens[SPA
    CE]opslaan[SPACE]([SPACE]saven[SPA
    CE])[CRSR-DOWN]"
40  printtab(7)f7$"[SPACE]stoppen[CRSR
    -DOWN]":printtab(7)"=============="
41  printtab(8)"maak[SPACE]u[SPACE]keu
    ze":printtab(7)"=============="
42  fork=0to16
43  forv=56184to56195:pokev,k:next
44  geti$:ifi$="[SPACE]"ori$=""then47
45  ifasc(i$)<133orasc(i$)>139then47
46  onasc(i$)-132goto51,124,194,252,16
    8,313,228
47  next:goto42
48  rem
49  rem gegevens team
50  rem
51  ifs(1,0,0)<>0then77
52  print"[SHIFT-CLR]"p$:printtab(10)"
    bowling[SPACE]team[SPACE]gegevens"
    :printp$"[CRSR-DOWN]"
53  input"naam[SPACE]team[10xSPACE]:[C
    TRL 9]";bt$
54  input"league[13xSPACE]:[CTRL-9]";l$
55  input"handicap[SPACE](ja/nee)[2xSP
    ACE]:[CTRL-9]";hc$
56  input"aantal[SPACE]games[SPACE](3/
    4)[SPACE]:[CTRL-9]";ag
57  input"seizoen[SPACE](j/j)[6xSPACE]
    :[CTRL-9]";jj$
58  input"[CRSR-DOWN]aantal[SPACE]spel
    ers[SPACE]team[SPACE](max[SPACE]10
    )";sp:print"[CRSR-DOWN]":gosub261
59  ifi$="j"then62
60  ifi$="n"thenprint"[HOME][3xCRSR-DO
    WN][CRSR-RIGHT]"
61  goto53
62  print"[SHIFT-CLR]"p$:printtab(15)"
    naam[SPACE]spelers":printp$"[CRSR-
    DOWN]"
63  fork=1tosp
64  input"naam[SPACE]speler[2xSPACE]:[
    CTRL-9]";n$(k)
65  next:print"[CRSR-DOWN]":gosub261
66  ifi$="j"then69
67  ifi$="n"thenprint"[HOME][3xCRSR-DO
    WN][CRSR-RIGHT]"
68  goto63
69  print"[SHIFT-CLR]"p$:printtab(8)"n
    aam[SPACE]-->[SPACE]computer[SPACE
    ]nummer":printp$"[CRSR-DOWN]"
70  fork=1tosp
71  printn$(k);:printtab(14);:input"[S
    PACE]:[CTRL-9]";cn:s(k,0,0)=cn
72  next:print"[CRSR-DOWN]":gosub261
73  ifi$="j"then29
74  ifi$="n"thenprint"[HOME][3xCRSR-DO
    WN][CRSR-RIGHT]"
75  goto70
76  rem
77  rem score invoer
78  rem
79  print"[SHIFT-CLR]"p$
80  print"score[SPACE]invoer[SPACE]voo
    r[SPACE]speler":printp$"[CRSR-DOWN]"
81  gosub267:gosub261:ifi$="n"then79
82  ifx=0then29
83  ifx=11thenprint"[CRSR-DOWN]alleen[
    SPACE]bij[SPACE]overzicht[SPACE]dr
    uk[SPACE]return":gosub393:goto79
84  gl=0:print"[SHIFT-CLR]score[SPACE]
    invoer[SPACE]speler:[SPACE]"n$(x)"
    [CRSR-DOWN]"
85  y=s(x,0,1):y=y+1:t=0:ts=0
86  input"datum[SPACE]:[CTRL-9]";d:pri
    nt"[CTRL-0]"
87  s(x,y,0)=d
88  forz=1toag
89  printz"[SPACE]game[SPACE]:[CTRL-9]
    ";:inputg
90  s(x,y,z)=g:t=t+g
91  ifg>glthengl=g
92  print"[CTRL-0]":nextz
93  print"[CRSR-UP][CTRL-0]"tab(20)"to
    taal[SPACE]:"t:print"[3xCRSR-DOWN]
    ":gosub261
94  ifi$="j"then96
95  ifi$="n"then84
96  s(x,y,5)=t
97  ts=s(x,y-1,6):ts=ts+t:s(x,y,6)=ts
98  dl=y*ag:rem deling aantal games
99  s(x,y,7)=int(ts/dl)
100 rem hcp berekening
101 ify<=7then107
102 hct=0
103 hct=s(x,y,6)-s(x,y-7,6)
104 sub=int(hct/21)
105 h=int((180-sub)*0.75)
106 goto108
107 h=int((180-s(x,y,7))*0.75)
108 ifh>45thenh=45:rem hcp max 45
109 ifh<0thenh=0 :rem hcp 0- 45
110 ifhc$="n"thenh=0
111 s(x,y,8)=h
112 rem hoogste serie
113 ift>s(x,0,2)thens(x,0,2)=t
114 rem berekening hoogste game
115 ifs(x,0,3)<glthens(x,0,3)=gl
116 rem y = aantal input
117 s(x,0,1)=y
118 print"[2xCRSR-DOWN]zijn[SPACE]er[S
    PACE]nog[SPACE]meer[SPACE]gegevens
    [SPACE][CRSR-DOWN]":gosub261
119 ifi$="n"then79
120 ifi$="j"then84
121 rem
122 rem score overzicht
123 rem
124 print"[SHIFT-CLR]"p$:printtab(12)"
    hcp[SPACE]en[SPACE]score[SPACE]ove
    rzicht":printp$"[CRSR-DOWN]":gosub267
125 gosub261:ifi$="n"then124
126 ifx=0then29
127 ifx=11then398
128 ifs(x,0,1)=0then124
```

```
129 print"[SHIFT-CLR][HOME][CRSR-RIGHT
    ][CTRL-9][COM-1]blz.0[SPACE][CTRL-
    2][SPACE]score[SPACE]overzicht[SPA
    CE]bowling[2xSPACE]"jj$"[SPACE][CT
    RL 0]":b=49
130 print"[COM-1]UDDDDDDDDDDDDDDDDDDDD
    DDDDDDDDDDDDDDDDDI"
131 print"G[CTRL-2]speler:[SPACE]"n$(x);
132 printtab(23)"cmpt[SPACE]nr[SPACE]:
    "s(x,0,0)tab(38)"[COM-1]H"
133 print"G"tab(38)"H"
134 print"G[CTRL-2]team[2xSPACE]:[SPAC
    E]"bt$tab(23)"h.game[2xSPACE]:"s(x
    ,0,3)tab(38)"[COM-1]H"
135 print"G"tab(38)"H"
136 print"G[CTRL-2]league:[SPACE]"l$ta
    b(23)"h.serie[SPACE]:"s(x,0,2)tab(
    38)"[COM-1]H"
137 print"JFFFFFFFFFFFFFFFFFFFFFFFFFFF
    FFFFFFFFFFK[CTRL-2]"
138 s%=1
139 fork=1toag:printtab(s%)k;
140 s%=s%+4
141 next k:k=0
142 printtab(17)"tot"tab(23)"sub"tab(3
    0)"gem"tab(35)"hcp":printp$
143 fory=1tos(x,0,1)
144 poke1029,b:rem b=blz
145 forz=1toag
146 ifs(x,y,z)>199thenprint"[COM-1]"s(
    x,y,z)"[CTRL-2][CRSR-LEFT]";:goto149
147 ifs(x,y,z)<100thenprint"[SPACE]"s(
    x,y,z)"[CRSR-LEFT]";:goto149
148 prints(x,y,z);:ifs(x,y,z)>99thenpr
    int"[CRSR-LEFT]";
149 next
150 printtab(16)s(x,y,5);
151 printtab(22)s(x,y,6);
152 printtab(29)s(x,y,7);:ifhc$="n"the
    nprinttab(35)"--":goto154
153 printtab(34)s(x,y,8):
154 k=k+1:ifk=10then156
155 nexty
156 k=0:print"[HOME][21xCRSR-DOWN]"
157 printf1$"[SPACE]volgend[SPACE]sche
    rm[SPACE]"f7$"score[SPACE]overzicht"
158 geti$:ifi$="[F1]"then161
159 ifi$="[F7]"then124
160 ifi$<>"[F1]"then158
161 forv=1424to1823:pokev,96:next:prin
    t"[HOME][9xCRSR-DOWN]":ify-1=s(x,0
    ,1)then164
162 ifs(x,o,1)=0then164
163 b=b+1:nexty
164 print"[4xCRSR-DOWN]volgend[SPACE]s
    cherm[SPACE]niet[SPACE]aanwezig":g
    oto156
165 rem
166 rem laden
167 rem
168 gosub298:print"[SHIFT-CLR]"p$:prin
    ttab(13)"gegevens[SPACE]laden":pri
    ntp$"[CRSR-DOWN]"
169 gosub301:print"[HOME][3xCRSR-DOWN]"
170 input"naam[SPACE]van[SPACE]bestand
    [SPACE]:[CTRL-9]";f$:print"[CTRL-0
    ][CRSR-DOWN]":f$="b."+f$
171 print"[HOME][7xCRSR-DOWN]";:gosub261
172 ifi$="n"then29
173 print"[CRSR-DOWN]druk[SPACE]return
    [SPACE]om[SPACE]bestand[SPACE]te[S
    PACE]laden":print"[CTRL-0][CRSR-UP]"
174 gosub 393
175 open2,8,2,"0:"+f$+",s,r":gosub286
176 input#2,d$,bt$,l$,hc$,jj$,ag$,sp$
177 ag=val(ag$):sp=val(sp$)
178 forx=1tosp:input#2,n$(x):next
179 forx=1tosp
180 forz=0to3
181 input#2,s(x,0,z)
182 nextz,x
183 forx=1tosp:k=s(x,0,1)
184 fory=1tok
185 forz=0to8
186 input#2,s(x,y,z):nextz,y,x
187 close2:close15
188 print"[3xCRSR-DOWN]laatste[SPACE]d
    atum[SPACE]van[SPACE]invoer[SPACE]
    was[SPACE]"d$:printtab(10)"druk[SP
    ACE]return[SPACE]"
189 gosub393
190 goto29
191 rem
192 rem gegevens wijzigen
193 rem
194 print"[SHIFT-CLR]"p$:printtab(10)"
    gegevens[SPACE]wijzigen":printp$"[
    CRSR-DOWN]"
195 printf2$"[SPACE]speler[SPACE]laten
    [SPACE]vervallen[CRSR-DOWN]"
196 printf4$"[SPACE]speler[SPACE]toevo
    egen[CRSR-DOWN]"
197 printf6$"[SPACE]naam[SPACE]team[CR
    SR-DOWN]"
198 printf8$"[SPACE]menu[CRSR-DOWN]"
199 ifsp=0thenprint"[3xCRSR-DOWN]geen[
    SPACE]bestand[SPACE]aanwezig...dru
    k[SPACE]f8"
200 getw$:ifw$="[SPACE]"orw$=""then200
201 ifasc(w$)<137orasc(w$)>140then200
202 onasc(w$)-136goto203,218,221,29
203 print"[SHIFT-CLR]"p$:printtab(10)"
    speler[SPACE]laten[SPACE]vervallen
    ":printp$
204 gosub267:gosub261:ifi$="n"then194
205 ifx=0then194
206 ifx=11thenprint"[CRSR-DOWN]alleen[
    SPACE]bij[SPACE]overzicht[SPACE]dr
    uk[SPACE]return":gosub393:goto203
207 n$(x)="":forz=0to4:s(x,0,z)=0:next
208 print"[CRSR-DOWN]wordt[SPACE]deze[
    SPACE]speler[SPACE]vervangen[SPACE
    ]j/n"
209 geti$:ifi$="[SPACE]"ori$=""then209
210 ifi$="n"then213
211 print:input"naam[SPACE]nieuwe[SPAC
    E]speler:[SPACE]";n$(x):input"comp
    uter[SPACE]nummer:[SPACE]";s(x,0,0)
212 goto194
213 ifx=spthen217
214 print"[CRSR-DOWN]even[SPACE]geduld
    [SPACE]a.u.b.....":fork=xtosp:n$(k
    )=n$(k+1):next
215 fork=xtosp:fory=0to52:forz=0to8
216 s(k,y,z)=s(k+1,y,z):nextz,y,k
217 sp=sp-1:goto194
```

```
218 print"[SHIFT-CLR]"p$:printtab(10)"
    speler[SPACE]toevoegen":printp$
219 print"[CRSR-DOWN]":input"naam[SPAC
    E]speler:[SPACE]";n$(sp+1):sp=sp+1
    :s(sp,0,1)=0
220 print"[CRSR-DOWN]":input"computer[
    SPACE]nummer:[SPACE]";s(sp,0,0):go
    to194
221 print"[SHIFT-CLR]"p$:printtab(10)"
    naam[SPACE]team[SPACE]wijzigen":pr
    intp$
222 print"[CRSR-DOWN]"tab(22)bt$:print
    "[2xCRSR-UP]"
223 input"naam[SPACE]team[10xSPACE]:[C
    TRL 9]";bt$
224 goto194
225 rem
226 rem saven
227 rem
228 gosub298:print"[SHIFT-CLR]"p$:prin
    ttab(12)"gegevens[SPACE]saven":pri
    ntp$"[CRSR-DOWN]"
229 gosub301:print"[HOME][3xCRSR-DOWN]"
230 printtab(20)bt$:print"[2xCRSR-UP]"
231 input"naam[SPACE]van[SPACE]bestand
    [SPACE]:[CTRL-9]";f$
232 printtab(19)d:print"[2xCRSR-UP]"
233 input"[CTRL-0]geef[SPACE]datum[7xS
    PACE]:[CTRL-9]";d$:print"[CTRL-0]"
    :f$="b."+f$
234 gosub261:ifi$="n"then29
235 print"[CRSR-DOWN]druk[SPACE]return
    [SPACE]om[SPACE]bestand[SPACE]te[S
    PACE]saven":print"[CTRL-0]"
236 gosub393
237 open2,8,2,"`0:"+f$+",s,w"
238 gosub286:ag$=str$(ag):sp$=str$(sp)
239 print#2,d$k$bt$k$l$k$hc$k$jj$k$ag$
    k$sp$
240 forx=1tosp:print#2,n$(x):next
241 forx=1tosp:forz=0to3
242 print#2,s(x,0,z):nextz,x
243 forx=1tosp:k=s(x,0,1)
244 fory=1tok
245 forz=0to8
246 print#2,s(x,y,z):nextz,y,x
247 close2:close15
248 goto29
249 rem
250 rem stoppen
251 rem
252 print"[SHIFT-CLR]"
253 print"[COM-1]"p$:printtab(17)"stop
    pen":printp$"[CRSR-DOWN]"
254 print"[2xCRSR-DOWN]heeft[SPACE]u[S
    PACE]de[SPACE]gegevens[SPACE]al[SP
    ACE]opgeborgen[CRSR-DOWN]":gosub261
255 ifi$="j"thenprint"[SHIFT-CLR][5xCR
    SR-DOWN][5xCRSR-RIGHT]tot[SPACE]de
    [SPACE]volgende[SPACE]keer........
    ":end
256 print"[SHIFT-CLR]":poke1707,45:pok
    e1708,62
257 poke55979,1:poke55980,1:goto30
258 rem
259 rem ja/nee
260 rem
261 i$="":printtab(15);:input"zeker[SP
    ACE]weten[SPACE](j/n):[2xSPACE]n[3
    xCRSR-LEFT]";i$
262 ifi$<>"n"andi$<>"j"then277
263 return
264 rem
265 rem keuze speler
266 rem
267 print"[HOME][4xCRSR-DOWN]voor[SPAC
    E]welke[SPACE]speler"
268 fork=0tosp:printk;:printn$(k):next
269 print"11[SPACE]"n$(11)
270 input"[CRSR-DOWN]geef[SPACE]nummer";x
271 ifx>sp and x<>11 then267
272 print"[CRSR-UP]"tab(17)"=[SPACE]"n
    $(x):print
273 return
274 rem
275 rem toon voor verkeerde aanslag
276 rem
277 poke54296,15 :rem volume
278 poke54277,50:rem a/v tijd 50
279 poke54278,248:rem a/v tijd 248
280 poke54273,50:poke54272,50:rem to
    on 50-50
281 poke54276,33:w=0
282 w=w+1:ifw<15then282
283 poke54276,0:poke54277,0:poke54278,0
284 print"[2xCRSR-UP]":goto261
285 rem
286 rem disk error reading
287 rem
288 open15,8,15:input#15,er,er$
289 ifer<20thenreturn
290 print"[CTRL-9][SPACE]disk[SPACE]er
    ror:[SPACE][CTRL-0]"er"[SPACE]"er$
    "[2xCRSR-DOWN]"
291 print"[CRSR-UP]druk[SPACE]"f1$"[2x
    SPACE]in[SPACE]voor[SPACE]menu":fo
    rw=1to100:next
292 geti$:print"[CRSR-UP][5xCRSR-RIGHT
    ][SPACE]f1[SPACE][CRSR-RIGHT]":for
    w=oto100:next
293 ifi$<>"[F1]"then291
294 close15:close2:goto29
295 rem
296 rem directory lezen bestanden
297 rem
298 print"[SHIFT-CLR][4xCRSR-DOWN][2xC
    RSR-RIGHT]plaats[SPACE]disk[SPACE]
    in[SPACE]drive[7xSPACE]druk[SPACE]
    return[SPACE]"
299 gosub393
300 return
301 print"[HOME][16xCRSR-DOWN]bestande
    n:[SPACE]";
302 open1,8,0,"$0"
303 get#1,b$:if st<>0then 309
304 ifb$<>chr$(34)then303
305 get#1,b$:ifb$<>chr$(34)thenc$=c$+b
    $:goto305
306 ifleft$(c$,2)="b."thenprinttab(12)
    mid$(c$,3,16):c$="":goto303
307 c$="":get#1,b$:ifb$=chr$(32)then303
308 ifst=0then303
309 close1:return
310 rem
311 rem gegevens printer
312 rem
```

```
313 x%=0:print"[SHIFT-CLR]druk[SPACE]r
    eturn[SPACE]als[SPACE]de[SPACE]pri
    nter[SPACE]aan[SPACE]staat?"
314 gosub 393
315 print"[SHIFT-CLR]"p$:printtab(17)"
    printen":printp$"[CRSR-DOWN]"
316 printf1$"[SPACE]team[SPACE]gegeven
    s[CRSR-DOWN]"
317 printf3$"[SPACE]alle[SPACE]spelers
    [CRSR-DOWN]"
318 printf5$"[SPACE]per[SPACE]speler[C
    RSR-DOWN]"
319 printf7$"[SPACE]menu[CRSR-DOWN]"
320 ifsp=0thenprint"[CRSR-DOWN]geen[SP
    ACE]bestand[SPACE]aanwezig[3xSPACE
    ]druk[SPACE]f7"
321 geti$:ifi$="[SPACE]"ori$=""then321
322 ifasc(i$)<133orasc(i$)>136then321
323 onasc(i$)-132goto324,337,339,29
324 print"[SHIFT-CLR]"p$:printtab(17)"
    team[SPACE]gegevens":printp$
325 open4,4:print#4,chr$(14)chr$(16)"1
    0gegevens[SPACE]bowling[SPACE]team
    :[SPACE]"bt$
326 print#4,chr$(15)chr$(13)
327 print#4,chr$(16)"10league[6xSPACE]
    :[SPACE]"l$chr$(13)
328 print#4,chr$(16)"10handicap[4xSPAC
    E]:[SPACE]"hc$chr$(13)
329 print#4,chr$(16)"10aantal[SPACE]ga
    mes:[SPACE]"agchr$(13)
330 print#4,chr$(16)"10seizoen[5xSPACE
    ]:[SPACE]"jj$chr$(13)chr$(13)
331 print#4,chr$(14)chr$(16)"10de[SPAC
    E]spelers"
332 print#4,chr$(15)chr$(10)
333 fork=1tosp
334 print#4,chr$(16)"10"s(k,0,0)chr$(1
    6)"16"n$(k)chr$(13):next
335 print#4,"":close4:goto315
336 rem
337 print"[SHIFT-CLR]"p$:printtab(10)"
    printen[SPACE]alle[SPACE]speler":p
    rintp$"[CRSR-DOWN]"
338 forx=1tosp:print"[CRSR-DOWN]"n$(x)
    :goto342
339 print"[SHIFT-CLR]"p$:printtab(10)"
    printen[SPACE]per[SPACE]speler":pr
    intp$"[CRSR-DOWN]"
340 gosub267:gosub261:x%=x
341 ifx=0then315
342 open4,4:print#4,chr$(14)chr$(16)"0
    5gegevens[SPACE]speler[SPACE]en[SP
    ACE]bowling[SPACE]score"
343 print#4,chr$(15)chr$(13)
344 print#4,chr$(16)"10naam[SPACE]team
    [3xSPACE]:[SPACE]"bt$
345 print#4,chr$(16)"10naam[SPACE]spel
    er[SPACE]:[SPACE]"n$(x);
346 print#4,chr$(16)"50computer[SPACE]
    nummer:"s(x,0,0)
347 print#4,chr$(16)"10league[6xSPACE]
    :[SPACE]"l$
348 print#4,chr$(16)"10handicap[4xSPAC
    E]:[SPACE]"hc$;
349 print#4,chr$(16)"50hoogste[SPACE]s
    erie[2xSPACE]:"s(x,0,2)
350 print#4,chr$(16)"10aantal[SPACE]ga
    mes:[SPACE]"ag
351 print#4,chr$(16)"10seizoen[5xSPACE
    ]:[SPACE]"jj$;
352 print#4,chr$(16)"50hoogste[SPACE]g
    ame[3xSPACE]:"s(x,0,3)
353 print#4,chr$(10)
354 print#4,chr$(16)"04datum";
355 print#4,chr$(16)"141";
356 print#4,chr$(16)"222";
357 print#4,chr$(16)"303";
358 ifs(x,y,4)=0then360
359 print#4,chr$(16)"384";
360 print#4,chr$(16)"45tot";
361 print#4,chr$(16)"53sub";
362 print#4,chr$(16)"62gem";
363 print#4,chr$(16)"69hcp"
364 forpp=1to80:print#4,"=";:next:print#4
365 fory=1tos(x,0,1):letdag$=str$(s(x,
    y,0))
366 iflen(dag$)=6thendag$="0"+mid$(dag
    $,2,6):goto368
367 dag$=mid$(dag$,2,6)
368 print#4,chr$(16)"04"dag$;
369 print#4,chr$(16)"12"s(x,y,1);:ifs(
    x,y,1)>=200thenprint#4,"[SPACE]*";
370 print#4,chr$(16)"20"s(x,y,2);:ifs(
    x,y,2)>=200thenprint#4,"[SPACE]*";
371 print#4,chr$(16)"28"s(x,y,3);:ifs(
    x,y,3)>=200thenprint#4,"[SPACE]*";
372 ifs(x,y,4)=0then374
373 print#4,chr$(16)"36"s(x,y,4);:ifs(
    x,y,4)>=200thenprint#4,"[SPACE]*";
374 print#4,chr$(16)"44"s(x,y,5);
375 print#4,chr$(16)"52"s(x,y,6);
376 print#4,chr$(16)"61"s(x,y,7);
377 ifhc$="nee"then380
378 ifs(x,y,8)=0then380
379 print#4,chr$(16)"68"s(x,y,8):goto381
380 print#4,""
381 nexty
382 forpp=1to80:print#4,"=";:next
383 print#4:close4
384 ifx%<>0then315
385 print"[CRSR-DOWN]stel[SPACE]printe
    r[SPACE]in[SPACE]voor[SPACE]de[SPA
    CE]volgende[SPACE]speler"
386 print"[CRSR-DOWN]druk[SPACE]return"
387 gosub393
388 nextx
389 goto315
390 rem
391 rem return chr$(13)
392 rem
393 geti$:ifi$<>chr$(13)then393
394 return
395 rem
396 rem overzicht alle spelers
397 rem
398 print"[SHIFT-CLR]"p$:printtab(8)"o
    verzicht[SPACE]alle[SPACE]spelers"
    :printp$"[CRSR-DOWN]"
399 print"naam"tab(15)"[SPACE]gem"tab(
    20)"hcp"tab(25)"gespeeld"
400 print"----"tab(15)"[SPACE]---"tab(
    20)"---"tab(25)"--------"
401 forx=1tosp:y=s(x,0,1):printn$(x)ta
    b(15)s(x,y,7)tab(20)s(x,y,8)tab(25
    )y:next
```

```
402 print"[CRSR-DOWN]druk[SPACE]return
    [SPACE]voor[SPACE]menu"
403 gosub 393
404 goto124
```

EINDE LISTING bowlen

Checksum Bowlen 64

REGEL 1	163	REGEL 82	215	REGEL 163	38	REGEL 244	172	REGEL 325	135
REGEL 2	8	REGEL 83	127	REGEL 164	215	REGEL 245	153	REGEL 326	242
REGEL 3	198	REGEL 84	200	REGEL 165	143	REGEL 246	28	REGEL 327	245
REGEL 4	175	REGEL 85	108	REGEL 166	243	REGEL 247	18	REGEL 328	185
REGEL 5	210	REGEL 86	184	REGEL 167	143	REGEL 248	244	REGEL 329	120
REGEL 6	247	REGEL 87	211	REGEL 168	233	REGEL 249	143	REGEL 330	35
REGEL 7	211	REGEL 88	234	REGEL 169	126	REGEL 250	184	REGEL 331	64
REGEL 8	235	REGEL 89	222	REGEL 170	149	REGEL 251	143	REGEL 332	239
REGEL 9	81	REGEL 90	133	REGEL 171	2	REGEL 252	112	REGEL 333	246
REGEL 10	250	REGEL 91	19	REGEL 172	78	REGEL 253	226	REGEL 334	51
REGEL 11	214	REGEL 92	133	REGEL 173	216	REGEL 254	131	REGEL 335	166
REGEL 12	121	REGEL 93	111	REGEL 174	44	REGEL 255	90	REGEL 336	143
REGEL 13	228	REGEL 94	78	REGEL 175	211	REGEL 256	218	REGEL 337	205
REGEL 14	187	REGEL 95	79	REGEL 176	86	REGEL 257	102	REGEL 338	162
REGEL 15	57	REGEL 96	232	REGEL 177	104	REGEL 258	143	REGEL 339	150
REGEL 16	203	REGEL 97	198	REGEL 178	246	REGEL 259	33	REGEL 340	77
REGEL 17	150	REGEL 98	78	REGEL 179	3	REGEL 260	143	REGEL 341	5
REGEL 18	143	REGEL 99	101	REGEL 180	148	REGEL 261	201	REGEL 342	74
REGEL 19	32	REGEL 100	68	REGEL 181	192	REGEL 262	67	REGEL 343	242
REGEL 20	143	REGEL 101	191	REGEL 182	96	REGEL 263	142	REGEL 344	84
REGEL 21	0	REGEL 102	193	REGEL 183	239	REGEL 264	143	REGEL 345	148
REGEL 22	42	REGEL 103	228	REGEL 184	172	REGEL 265	222	REGEL 346	81
REGEL 23	138	REGEL 104	145	REGEL 185	153	REGEL 266	143	REGEL 347	121
REGEL 24	46	REGEL 105	245	REGEL 186	8	REGEL 267	189	REGEL 348	120
REGEL 25	203	REGEL 106	34	REGEL 187	18	REGEL 268	235	REGEL 349	161
REGEL 26	143	REGEL 107	239	REGEL 188	254	REGEL 269	100	REGEL 350	252
REGEL 27	52	REGEL 108	234	REGEL 189	44	REGEL 270	88	REGEL 351	102
REGEL 28	143	REGEL 109	241	REGEL 190	244	REGEL 271	74	REGEL 352	68
REGEL 29	131	REGEL 110	79	REGEL 191	143	REGEL 272	17	REGEL 353	113
REGEL 30	170	REGEL 111	223	REGEL 192	74	REGEL 273	142	REGEL 354	213
REGEL 31	143	REGEL 112	32	REGEL 193	143	REGEL 274	143	REGEL 355	140
REGEL 32	86	REGEL 113	169	REGEL 194	127	REGEL 275	169	REGEL 356	140
REGEL 33	9	REGEL 114	156	REGEL 195	120	REGEL 276	143	REGEL 357	140
REGEL 34	49	REGEL 115	245	REGEL 196	3	REGEL 277	212	REGEL 358	142
REGEL 35	21	REGEL 116	102	REGEL 197	210	REGEL 278	80	REGEL 359	149
REGEL 36	54	REGEL 117	192	REGEL 198	197	REGEL 279	195	REGEL 360	86
REGEL 37	233	REGEL 118	163	REGEL 199	123	REGEL 280	147	REGEL 361	72
REGEL 38	75	REGEL 119	83	REGEL 200	172	REGEL 281	164	REGEL 362	55
REGEL 39	193	REGEL 120	75	REGEL 201	44	REGEL 282	179	REGEL 363	5
REGEL 40	43	REGEL 121	143	REGEL 202	165	REGEL 283	104	REGEL 364	192
REGEL 41	68	REGEL 122	201	REGEL 203	93	REGEL 284	91	REGEL 365	108
REGEL 42	185	REGEL 123	143	REGEL 204	71	REGEL 285	143	REGEL 366	13
REGEL 43	153	REGEL 124	212	REGEL 205	10	REGEL 286	62	REGEL 367	109
REGEL 44	91	REGEL 125	218	REGEL 206	164	REGEL 287	143	REGEL 368	74
REGEL 45	237	REGEL 126	215	REGEL 207	150	REGEL 288	201	REGEL 369	23
REGEL 46	237	REGEL 127	66	REGEL 208	228	REGEL 289	108	REGEL 370	24
REGEL 47	171	REGEL 128	96	REGEL 209	139	REGEL 290	108	REGEL 371	34
REGEL 48	143	REGEL 129	243	REGEL 210	121	REGEL 291	123	REGEL 372	147
REGEL 49	10	REGEL 130	80	REGEL 211	80	REGEL 292	223	REGEL 373	35
REGEL 50	143	REGEL 131	4	REGEL 212	39	REGEL 293	104	REGEL 374	64
REGEL 51	193	REGEL 132	148	REGEL 213	121	REGEL 294	64	REGEL 375	64
REGEL 52	81	REGEL 133	231	REGEL 214	225	REGEL 295	143	REGEL 376	65
REGEL 53	78	REGEL 134	126	REGEL 215	241	REGEL 296	86	REGEL 377	74
REGEL 54	115	REGEL 135	231	REGEL 216	147	REGEL 297	143	REGEL 378	148
REGEL 55	26	REGEL 136	29	REGEL 217	53	REGEL 298	5	REGEL 379	109
REGEL 56	221	REGEL 137	21	REGEL 218	230	REGEL 299	44	REGEL 380	60
REGEL 57	57	REGEL 138	91	REGEL 219	184	REGEL 300	142	REGEL 381	219
REGEL 58	206	REGEL 139	120	REGEL 220	9	REGEL 301	9	REGEL 382	186
REGEL 59	71	REGEL 140	128	REGEL 221	26	REGEL 302	84	REGEL 383	218
REGEL 60	35	REGEL 141	52	REGEL 222	17	REGEL 303	202	REGEL 384	220
REGEL 61	241	REGEL 142	77	REGEL 223	78	REGEL 304	17	REGEL 385	207
REGEL 62	4	REGEL 143	22	REGEL 224	39	REGEL 305	39	REGEL 386	4
REGEL 63	246	REGEL 144	1	REGEL 225	143	REGEL 306	162	REGEL 387	44
REGEL 64	70	REGEL 145	234	REGEL 226	12	REGEL 307	181	REGEL 388	218
REGEL 65	10	REGEL 146	116	REGEL 227	143	REGEL 308	81	REGEL 389	34
REGEL 66	78	REGEL 147	222	REGEL 228	1	REGEL 309	153	REGEL 390	143
REGEL 67	35	REGEL 148	38	REGEL 229	126	REGEL 310	143	REGEL 391	37
REGEL 68	242	REGEL 149	130	REGEL 230	186	REGEL 311	7	REGEL 392	143
REGEL 69	147	REGEL 150	233	REGEL 231	189	REGEL 312	143	REGEL 393	102
REGEL 70	246	REGEL 151	231	REGEL 232	76	REGEL 313	56	REGEL 394	142
REGEL 71	96	REGEL 152	22	REGEL 233	163	REGEL 314	44	REGEL 395	143
REGEL 72	10	REGEL 153	235	REGEL 234	174	REGEL 315	235	REGEL 396	137
REGEL 73	74	REGEL 154	137	REGEL 235	96	REGEL 316	4	REGEL 397	143
REGEL 74	35	REGEL 155	219	REGEL 236	44	REGEL 317	199	REGEL 398	149
REGEL 75	240	REGEL 156	188	REGEL 237	177	REGEL 318	63	REGEL 399	86
REGEL 76	143	REGEL 157	104	REGEL 238	205	REGEL 319	196	REGEL 400	102
REGEL 77	222	REGEL 158	250	REGEL 239	252	REGEL 320	206	REGEL 401	92
REGEL 78	143	REGEL 159	180	REGEL 240	10	REGEL 321	134	REGEL 402	127
REGEL 79	228	REGEL 160	106	REGEL 241	209	REGEL 322	21	REGEL 403	44
REGEL 80	217	REGEL 161	170	REGEL 242	110	REGEL 323	163	REGEL 404	32
REGEL 81	25	REGEL 162	131	REGEL 243	239	REGEL 324	241		

Spreadsheet 64

De Heer O.R. Vos uit Amersfoort volgde een cursus Lotus 1 2 3. Thuis achter zijn trouwe Commodore 64 heeft hij zijn pas verworven kennis gekombineerd met reeds bestaande programmeer kennis gekoppeld tot het volgende programma. Een uitleg hoe het programma gebruikt moet worden is in het programma opgenomen. Deze wordt verkregen door het indrukken van de ?.

```
5    print"[SHIFT-CLR]":poke53280,0:pok
     e53281,0:dimcm$(2600):poke650,0
10   rem
15   l1$="[CTRL-9][4xSPACE][CTRL-0][10x
     SHIFT-*][SHIFT-+][10xSHIFT-*][SHIF
     T +][10xSHIFT-*][COM-W][COM-G]"
20   l2$="[4xCRSR-RIGHT][10xSPACE]B[10x
     SPACE]B[10xSPACE]B[COM-G]"
25   w$="[10xSPACE][SHIFT--][11xCRSR-LE
     FT][CTRL-4]"
30   x$="[CTRL-9][37xSPACE][CTRL-0][COM G]"
35   y$="[39xSPACE]"
40   z$="[CTRL-9]abcdefghijklmnopqrstuv
     wxyz[CTRL-0]"
45   r1=1:kl=2:r2=1:cd=1
50   print"[HOME][4xCRSR-DOWN][CTRL-6]";
55   print"[CTRL-9][9xSPACE]"mid$(z$,kl
     ,1)"[9xSPACE]"mid$(z$,kl+1,1);
60   print"[11xSPACE]"mid$(z$,kl+2,1)"[5xSP
     ACE][CTRL-0][COM-H]":printl1$:d=r2
65   fori=r1tor1+7
70   print"[CTRL-9][4xSPACE][4xCRSR-LEF
     T][CTRL-6]"i"[CTRL-0]":print"[CRSR
     -UP][4xCRSR-RIGHT]"w$;left$(cm$(d)
     ,10)"[CTRL-6]";
75   printtab(15)w$;left$(cm$(d+100),10
     )"[CTRL-6]";
80   printtab(26)w$;left$(cm$(d+200),10
     )"[CTRL-6]"tab(37)"[COM-H]"
85   d=d+1:printl1$:next
90   print"[CTRL-9][6xSPACE](c)[SPACE]1
     987[2xSPACE]o.r.vos[SPACE]-[SPACE]
     software[3xSPACE][CTRL-0][COM-G]"
95   printtab(14)"[CTRL-9][CTRL-4][PIJL
     LINKS][SPACE]=[SPACE]escape[CTRL-6
     ][CTRL-0][3xSPACE]?[SPACE]=[SPACE]
     help[HOME]"
100  print"[HOME][CTRL-0][SPACE][CTRL-4
     ]commando's[SPACE][CTRL-9][SPACE]n
     [SPACE]g[SPACE]h[SPACE]w[SPACE]f[S
     PACE]c[SPACE]s[SPACE]l[SPACE]z[SPA
     CE]m[SPACE][CTRL-6][CTRL-0][SPACE]";y$
105  rem
110  printy$"[CRSR-UP][CTRL-6][SPACE]co
     mmand:";
115  gosub480
120  ifa$="n"then920
125  ifa$="g"thent=1:fo=0:goto195
130  ifa$="h"then45
135  ifa$="w"then510
140  ifa$="f"then310
145  ifa$="c"then525
150  ifa$="m"then520
155  ifa$="s"then665
160  ifa$="l"then540
165  ifa$="z"then970
170  ifa$="[CRSR-RIGHT]"andkl<25thenkl=
     kl+1:r2=r2+100:goto50
175  ifa$="[CRSR-LEFT]"andkl>2thenkl=kl
     -1:r2=r2-100:goto50
180  ifa$="[CRSR-DOWN]"andr1<93thenr1=r
     1+1:r2=r2+1:goto50
185  ifa$="[CRSR-UP]"andr1>1thenr1=r1-1
     :r2=r2-1:goto50
190  goto115
195  print"[HOME][2xCRSR-DOWN][CTRL-6]g
     oto[SPACE]cel:";
200  c=0:r=0:o=0:v$=""
205  gosub480
210  ifa$=chr$(13)then205
215  c=asc(a$)-65:ifc=0then225
220  r=100
225  ifc<0orc>26then200
230  ift=1thent=0:goto240
235  return
240  printa$;
245  gosub480
250  ifa$=chr$(13)then275
255  b=asc(a$):ifb<48orb>57then245
260  v$=v$+a$:o=val(v$):ifo<0oro>100the
     nprint"[CTRL-0][3xCRSR-LEFT][3xSPA
     CE][3xCRSR-LEFT]";:t=1:goto200
265  printa$;
270  goto245
275  cel=(c*r)+o:ifcel>cdthencd=cel
280  iffo=1thenreturn
285  print"[HOME][2xCRSR-DOWN][CTRL-9]e
     nter[CTRL-0][8xSPACE][7xCRSR-LEFT]
     ";:poke19,1:inputcm$(cel):poke19,0
290  cm$(cel)=cm$(cel)+"[10xSPACE]"
295  goto50
300  print"[HOME][CTRL-9][CTRL-2][2xSPA
     CE]laatst[SPACE]ingevoerde[SPACE]c
     el[SPACE]is[SPACE]leeg[6xSPACE][CT
     RL 0][CTRL-6]":gosub710:goto310
305  print"[HOME][CTRL-9][CTRL-2][2xSPA
     CE]cel[SPACE]bevat[SPACE][CTRL-3]g
     een[CTRL-2][SPACE]rekenkundig[SPAC
     E]getal[3xSPACE][CTRL-0][CTRL-6]":
     gosub710
310  print"[HOME][CTRL-9][CTRL-4]u[SPAC
     E]kunt[SPACE]alleen[3xSPACE]+[SPAC
     E]-[SPACE]*[SPACE]~[SPACE]/[3xSPAC
     E]gebruiken[CTRL-0][CTRL-6][15xSPACE]"
315  print"[HOME][2xCRSR-DOWN][CTRL-9]f
     ormule:[CTRL-0][10xSPACE][10xCRSR-
     LEFT]";:q=0:uit=0:do=0:t=0
320  gosub200:printa$;
325  gosub480
330  ifa$="+"thenprinta$;:goto380
335  ifa$="-"thenprinta$;:goto380
340  ifa$="*"thenprinta$;:goto380
345  ifa$="/"thenprinta$;:goto380
350  ifa$="~"thenprinta$;:goto380
355  ifa$="="thenprinta$;:uit=1:goto380
360  b=asc(a$):ifb<48orb>57then325
365  v$=v$+a$:o=val(v$):ifo<0oro>100the
     nprint"[CTRL-0][3xCRSR-LEFT][3xSPA
     CE][3xCRSR-LEFT]";:fo=1:gosub200
370  printa$;
375  goto325
380  ifdo=1thend$=r$:r$=a$:goto390
385  r$=a$
390  q=q+1
```

```
395  cel=(c*r)+o
400  ifcm$(cel)=""then300
405  ifasc(cm$(cel))=<46orasc(cm$(cel))
     =>58then305
410  ifq=1thenf(1)=val(cm$(cel)):v$="":
     do=1:goto320
415  ifq=2thenf(2)=val(cm$(cel)):v$=""
420  ifd$="+"thenuit$=str$(f(1)+f(2))
425  ifd$="-"thenuit$=str$(f(1)-f(2))
430  ifd$="*"thenuit$=str$(f(1)*f(2))
435  ifd$="/"thenuit$=str$(f(1)/f(2))
440  ifd$="[KWADRAATPIJL]"thenuit$=str$
     (f(1)[KWADRAATPIJL]f(2))
445  f(1)=val(uit$):q=1
450  ifuit=1thenuit=0:q=0:goto460
455  goto320
460  fo=1:t=1:gosub200:fo=0:t=0
465  cm$(cel)=str$(f(1)):f(1)=0
470  ha=val(cm$(cel))
475  cm$(cel)=mid$(str$(int((ha+0.51*10
     [KWADRAATPIJL]-2)*10[KWADRAATPIJL]2
     )*10[KWADRAATPIJL]-2),2,9):goto50
480  poke204,0:poke198,0:wait198,1:geta$
485  poke207,0:poke204,1
490  ifa$=chr$(20)then480
495  ifa$="?"then720
500  ifa$="[PIJL LINKS]"then45
505  return
510  print"[HOME]"y$y$y$y$"[HOME][CTRL9][C
     TRL-4][SPACE]wissen[SPACE][CTRL 0]"
515  print"welke[SPACE]cel:[CTRL-6]";:f
     o=1:t=1:gosub200:cm$(cel)="":fo=0:
     goto50
520  print"[HOME]"y$y$y$y$"[HOME][CTRL
     9][CTRL-4][SPACE]verplaatsen[SPACE
     ][CTRL-0]":v=1:goto530
525  print"[HOME]"y$y$y$y$"[HOME][CTRL
     9][CTRL-4][SPACE]copy[SPACE][CTRL0]"
530  print"welke[SPACE]cel:[CTRL-6]";:f
     o=1:t=1:gosub200:c$=cm$(cel):ifv=1
     thencm$(cel)="":v=0
535  print"[SPACE][CTRL-4]naar[SPACE]ce
     l:[CTRL-6]";:fo=1:t=1:gosub200:cm$
     (cel)=c$:fo=0:t=0:goto50
540  print"[SHIFT-CLR][CTRL-9][CTRL-4][
     SPACE]scanning[SPACE][CTRL-0]"
545  print"file's[SPACE]on[SPACE]disk[S
     PACE]:[CRSR-DOWN][7xCRSR-LEFT][CTR
     L 6]bloks[3xSPACE]prog.naam[CTRL-4
     ]":d$="":a$="":b$="":c$=""
550  open1,8,0,"$0":get#1,a$,b$
555  get#1,a$,b$
560  get#1,a$,b$
565  m=0
570  ifa$<>""thenm=asc(a$)
575  ifb$<>""thenm=m+asc(b$)*256
580  d$=d$+"[CTRL-9][SPACE]"+mid$(str$(
     m),2)+"[SPACE][CTRL-0][SPACE]--[SPACE]"
585  get#1,b$:ifst<>0thenclose1:print"[CR
     SR-DOWN]terug[SPACE]naar[SPACE]menu[3x
     SPACE]j/n[SPACE]";:gosub480:goto900
590  ifb$<>chr$(34)then585
595  get#1,b$:ifb$<>chr$(34)thend$=d$+b
     $:goto595
600  get#1,b$:ifb$=chr$(32)then600
605  c$=""
610  c$=c$+b$:get#1,b$:ifb$<>""then610
615  ifright$(d$,1)="[PIJL LINKS]"thenp
     rinttab(12)d$
620  d$="":goto555
625  print:print"[CTRL-9][SPACE]laden[S
     PACE][CTRL-0]":input"[CTRL-4]file-
     naam[SPACE]:[CTRL-6]";fl$
630  open2,8,2,"`0:"+fl$+"[SPACE][PIJL
     LINKS],s,r"
635  input#2,cd
640  forcel=1tocd
645  input#2,cm$(cel)
650  ifcm$(cel)="."thencm$(cel)=""
655  next
660  close2:print"[SHIFT-CLR]":goto45
665  print"[HOME]"y$y$y$y$"[HOME][CTRL
     9][CTRL-4][SPACE]saven[SPACE][CTRL 0]"
670  input"[CTRL-4]file-naam[SPACE]:[CT
     RL 6]";fl$
675  open2,8,2,"`0:"+fl$+"[SPACE][PIJL
     LINKS],s,w"
680  print#2,cd
685  forcel=1tocd
690  ifcm$(cel)=""thenprint#2,".":goto700
695  print#2,cm$(cel)
700  next
705  goto660
710  forw=0to3000:next:return
715  next
720  print"[HOME][4xCRSR-DOWN]"
725  print"[4xCRSR-RIGHT][32xSPACE]"
730  print"[4xCRSR-RIGHT][SPACE]n-toets
     [SPACE]=[SPACE]wis[SPACE]gehele[SP
     ACE]sheet[5xSPACE]"
735  print"[4xCRSR-RIGHT][SPACE]g-toets
     [SPACE]=[SPACE]ga[SPACE]naar[SPACE
     ]cel[SPACE]x[8xSPACE]"
740  print"[4xCRSR-RIGHT][SPACE]h-toets
     [SPACE]=[SPACE]terug[SPACE]naar[SP
     ACE]cel[SPACE]a1[4xSPACE]"
745  print"[4xCRSR-RIGHT][SPACE]w-toets
     [SPACE]=[SPACE]wis[SPACE]een[SPACE
     ]cel[10xSPACE]"
750  print"[4xCRSR-RIGHT][SPACE]m-toets
     [SPACE]=[SPACE]verplaats[SPACE]1[S
     PACE]cel[SPACE]naar[SPACE]"
755  print"[4xCRSR-RIGHT][11xSPACE]een[
     SPACE]andere[SPACE]cel[7xSPACE]"
760  print"[4xCRSR-RIGHT][SPACE]c-toets
     [SPACE]=[SPACE]copieer[SPACE]1[SPA
     CE]cel[SPACE]naar[3xSPACE]"
765  print"[4xCRSR-RIGHT][11xSPACE]een[
     SPACE]andere[SPACE]cel[7xSPACE]"
770  print"[4xCRSR-RIGHT][SPACE]s-toets[SPA
     CE]=[SPACE]bewaar[SPACE]sheet[9xSPACE]"
775  print"[4xCRSR-RIGHT][SPACE]l-toets[SP
     ACE]=[SPACE]laad[SPACE]sheet[11xSPACE]"
780  print"[4xCRSR-RIGHT][32xSPACE]"
785  print"[4xCRSR-RIGHT][32xSPACE]"
790  print"[4xCRSR-RIGHT][SPACE]cursor-
     toetsen[SPACE]=[SPACE]wandelen[SPA
     CE]door[SPACE]"
795  print"[4xCRSR-RIGHT][18xSPACE]de[S
     PACE]sheet[6xSPACE]"
800  print"[4xCRSR-RIGHT][2xSPACE][CTRL 9][
     CTRL-4][SPACE]druk[3xSPACE]spatie[2x
     SPACE][CTRL-0][CTRL-6][14xSPACE]"
805  print"[4xCRSR-RIGHT][32xSPACE][HOM
     E][2xCRSR-DOWN][8xCRSR-RIGHT]";
810  gosub480
```

```
815  print"[HOME][4xCRSR-DOWN]"
820  print"[4xCRSR-RIGHT][32xSPACE]"
825  print"[4xCRSR-RIGHT][SPACE]z-toets
     [SPACE]=[SPACE]zoek[SPACE]een[SPAC
     E]gegeven[5xSPACE]"
830  print"[4xCRSR-RIGHT][32xSPACE]"
835  print"[4xCRSR-RIGHT][SPACE]f-toets
     [SPACE]=[SPACE]b.v.[SPACE]a1*b4-c2
     +h6/d3=[SPACE]"
840  print"[4xCRSR-RIGHT][11xSPACE]dus[
     SPACE][CTRL-9][CTRL-2]eerst[CTRL-0
     ][CTRL-6][SPACE]formuleren[SPACE]"
845  print"[4xCRSR-RIGHT][11xSPACE]dan[
     SPACE]aangeven[SPACE]in[SPACE]welke"
850  print"[4xCRSR-RIGHT][11xSPACE]cel[
     2xSPACE]de[SPACE]uitkomst[5xSPACE]"
855  print"[4xCRSR-RIGHT][11xSPACE]komt
     [SPACE]te[SPACE]staan[8xSPACE]"
860  print"[4xCRSR-RIGHT][32xSPACE]"
865  print"[4xCRSR-RIGHT][SPACE][PIJL L
     INKS]-toets[SPACE]=[SPACE]escape[SPA
     CE]op[SPACE]elk[SPACE]moment[SPACE]"
870  print"[4xCRSR-RIGHT][11xSPACE]behalve[
     SPACE]by[SPACE]enter[SPACE]en[2xSPACE]"
875  print"[4xCRSR-RIGHT][11xSPACE]zoek
     en[15xSPACE]"
880  print"[4xCRSR-RIGHT][32xSPACE]"
885  print"[4xCRSR-RIGHT][32xSPACE]"
890  print"[4xCRSR-RIGHT][32xSPACE][HOM
     E][2xCRSR-DOWN][8xCRSR-RIGHT]";
895  gosub480:goto45
900  ifa$="n"then625
905  ifa$="j"thenprint"[SHIFT-CLR]":goto45
910  goto855
915  return
920  print"[HOME][11xCRSR-DOWN][12xCRSR
     -RIGHT][CTRL-9][15xSPACE][CTRL-0]"
925  print"[12xCRSR-RIGHT][CTRL-9][SPACE]l
     ege[2xSPACE]sheet[SPACE]![SPACE][CTRL-0]"
930  print"[12xCRSR-RIGHT][CTRL-9][15xS
     PACE][CTRL-0]"
935  print"[12xCRSR-RIGHT][CTRL-9][SPACE]z
     eker[SPACE]weten[SPACE]![SPACE][CTRL-0]"
940  print"[12xCRSR-RIGHT][CTRL-9][4xSP
     ACE](j/n)[6xSPACE][CTRL-0]"
945  print"[12xCRSR-RIGHT][CTRL-9][15xS
     PACE][CTRL-0]"
950  geta$:ifa$=""then950
955  ifa$="n"then45
960  ifa$="j"thenrun
965  goto950
970  print"[HOME][2xCRSR-DOWN]zoek[SPAC
     E]gegeven[SPACE]:[CTRL-4]";:
975  poke19,1:inputzk$:poke19,0
980  bl=len(zk$)
985  fori=1tocd:print"[HOME][3xCRSR-DOWN]";
990  ifzk$=left$(cm$(i),bl)then1005
995  next
1000 print"[CRSR-UP]komt[SPACE]niet[SPA
     CE]in[SPACE]het[SPACE]bestand[SPAC
     E]voor!":gosub710:goto50
1005 ifi>100thenpc=int(i/100)+1:goto1015
1010 pc=1
1015 a=int(i-((pc-1)*100))
1020 print"[CRSR-UP]gegeven[SPACE]staat[SP
     ACE]in[SPACE]cel[SPACE]"chr$(pc+64);a
1025 gosub710:goto50
```

EINDE LISTING spreadsheet 64

Checksum spreadsheet 64

REGEL 5	36	REGEL 210	92	REGEL 415	123	REGEL 620	192	REGEL 825	182
REGEL 10	0	REGEL 215	177	REGEL 420	211	REGEL 625	13	REGEL 830	81
REGEL 15	201	REGEL 220	149	REGEL 425	214	REGEL 630	87	REGEL 835	196
REGEL 20	247	REGEL 225	246	REGEL 430	212	REGEL 635	105	REGEL 840	134
REGEL 25	172	REGEL 230	248	REGEL 435	218	REGEL 640	99	REGEL 845	120
REGEL 30	187	REGEL 235	142	REGEL 440	10	REGEL 645	187	REGEL 850	46
REGEL 35	115	REGEL 240	57	REGEL 445	20	REGEL 650	254	REGEL 855	156
REGEL 40	247	REGEL 245	41	REGEL 450	165	REGEL 655	130	REGEL 860	81
REGEL 45	96	REGEL 250	99	REGEL 455	30	REGEL 660	168	REGEL 865	165
REGEL 50	141	REGEL 255	231	REGEL 460	99	REGEL 665	251	REGEL 870	244
REGEL 55	187	REGEL 260	125	REGEL 465	76	REGEL 670	27	REGEL 875	29
REGEL 60	36	REGEL 265	57	REGEL 470	42	REGEL 675	92	REGEL 880	81
REGEL 65	7	REGEL 270	36	REGEL 475	17	REGEL 680	125	REGEL 885	81
REGEL 70	254	REGEL 275	150	REGEL 480	99	REGEL 685	99	REGEL 890	169
REGEL 75	72	REGEL 280	56	REGEL 485	80	REGEL 690	195	REGEL 895	85
REGEL 80	62	REGEL 285	129	REGEL 490	95	REGEL 695	207	REGEL 900	120
REGEL 85	69	REGEL 290	82	REGEL 495	101	REGEL 700	130	REGEL 905	115
REGEL 90	32	REGEL 295	238	REGEL 500	85	REGEL 705	37	REGEL 910	43
REGEL 95	116	REGEL 300	171	REGEL 505	142	REGEL 710	165	REGEL 915	142
REGEL 100	102	REGEL 305	64	REGEL 510	87	REGEL 715	130	REGEL 920	171
REGEL 105	143	REGEL 310	68	REGEL 515	26	REGEL 720	52	REGEL 925	148
REGEL 110	125	REGEL 315	1	REGEL 520	145	REGEL 725	81	REGEL 930	221
REGEL 115	41	REGEL 320	146	REGEL 525	185	REGEL 730	174	REGEL 935	2
REGEL 120	118	REGEL 325	41	REGEL 530	74	REGEL 735	103	REGEL 940	245
REGEL 125	30	REGEL 330	79	REGEL 535	246	REGEL 740	129	REGEL 945	221
REGEL 130	62	REGEL 335	81	REGEL 540	4	REGEL 745	64	REGEL 950	107
REGEL 135	122	REGEL 340	78	REGEL 545	234	REGEL 750	112	REGEL 955	68
REGEL 140	103	REGEL 345	83	REGEL 550	166	REGEL 755	172	REGEL 960	97
REGEL 145	108	REGEL 350	130	REGEL 555	24	REGEL 760	187	REGEL 965	39
REGEL 150	113	REGEL 355	112	REGEL 560	24	REGEL 765	172	REGEL 970	154
REGEL 155	129	REGEL 360	230	REGEL 565	47	REGEL 770	200	REGEL 975	125
REGEL 160	114	REGEL 365	194	REGEL 570	186	REGEL 775	33	REGEL 980	29
REGEL 165	135	REGEL 370	57	REGEL 575	252	REGEL 780	81	REGEL 985	112
REGEL 170	28	REGEL 375	35	REGEL 580	225	REGEL 785	81	REGEL 990	148
REGEL 175	103	REGEL 380	94	REGEL 585	198	REGEL 790	61	REGEL 995	130
REGEL 180	121	REGEL 385	141	REGEL 590	29	REGEL 795	83	REGEL 1000	64
REGEL 185	190	REGEL 390	47	REGEL 595	52	REGEL 800	174	REGEL 1005	244
REGEL 190	32	REGEL 395	17	REGEL 600	30	REGEL 805	169	REGEL 1010	118
REGEL 195	178	REGEL 400	148	REGEL 605	93	REGEL 810	41	REGEL 1015	59
REGEL 200	168	REGEL 405	249	REGEL 610	98	REGEL 815	52	REGEL 1020	146
REGEL 205	41	REGEL 410	129	REGEL 615	150	REGEL 820	81	REGEL 1025	76

PRINT OUT C-128 met Music maker

Musicmaker

Eindelijk, na een lange periode weer eens een muziek programma. Het is gemaakt door C. Duiker uit Zoetermeer. Het is zeker geen eenvoudig programma, maar na een tijdje stoeien blijken er vele dingen mee mogelijk. In het programma is een duidelijke gebruiks aanwijzing opgenomen. Deze handleiding kan ook worden uitgeprint.

```
10    with 1
20    poke775,255
30    trap 1970
40    scnclr:print:print"[2xSPACE]gebrui
      ksaanwijzing[SPACE]<j/n>"
50    getkey ga$
60    if ga$="j"thengoto1390
70    if ga$="n"thengoto90
80    goto50
90    key 1,"|"
100   key 3,")"
110   key 5,"!":key 7,"%"
120   tempo 255
130   color 1,1
140   color 4,1
150   color 0,2
160   c=0
170   dim vo$(100)
180   graphic1,1
190   a=10:b=30
200   t=2
210   fori=1to15
220   box 1,a,0,b,50
230   a=a+20:b=b+20
240   next i
250   box 1,23,0,37,30,0,1
260   box 1,43,0,57,30,0,1
270   box 1,83,0,97,30,0,1
280   box 1,103,0,117,30,0,1
290   box 1,123,0,137,30,0,1
300   box 1,163,0,177,30,0,1
310   box 1,183,0,197,30,0,1
320   box 1,223,0,237,30,0,1
330   box 1,243,0,257,30,0,1
340   box 1,263,0,277,30,0,1
350   box 1,303,0,310,30,0,1
360   color 1,7
370   char 1,2,7,"o0
380   char 1,2,5,"c"
390   char 1,4,5,"d"
400   char 1,7,5,"e"
410   char 1,9,5,"f"
420   char 1,12,5,"g"
430   char 1,14,5,"a"
440   char 1,17,5,"b"
450   char 1,19,7,"o1"
460   char 1,19,5,"c"
470   char 1,22,5,"d"
480   char 1,25,5,"e"
490   char 1,27,5,"f"
500   char 1,30,5,"g"
510   char 1,32,5,"a"
520   char 1,35,5,"b"
530   char 1,37,5,"c"
540   char 1,37,7,"o2
550   color 1,1
560   circle 1,150,150,20,20
570   paint 1,150,150
580   with 2
590   draw 1,170,150to170,90
600   draw 1,170,90to178,99
610   draw 1,170,95to178,104
620   with 1
630   color 1,3
640   char 1,1,23,"music[SPACE]maker[SPA
      CE]v7.0[3xSPACE]by[SPACE]c.duijker."
650   char 1,1,24,"[9xSPACE]powersoft[SP
      ACE]inc."
660   getkey a$
670   if a$="#"ora$="$"ora$="."ora$="o"t
      hengoto840
680   if a$="u"ora$="v"ora$="w"ora$="h"t
      hengoto840
690   if a$="q"ora$="i"ora$="s"ora$="r"t
      hengoto840
700   if a$="m"ora$="1"ora$="2"ora$="3"t
      hengoto840
710   if a$="4"ora$="5"ora$="6"thengoto840
720   if a$="0"thengoto840
730   if a$="*"thengoto940
740   if a$="t"thengoto840
750   if a$="[PIJL LINKS]"thengoto960
760   if a$="|"thengoto990
770   if a$=")"thengoto1030
780   if a$="!"thengoto1070
790   if a$="%"thengoto1190
800   if a$=":"thengoto1320
810   if a$="~"thengoto1810
820   if a$="/"thengoto1870
830   if a$<"a"ora$>"g"thengoto660
840   r=r+1:if r=38 then r=r+1
850   if r=39 then goto890
860   char 1,r,10,a$
870   b$=b$+a$
880   goto660
890   c=c+1
900   vo$(c)=b$
910   r=0
920   char 1,1,10,"[40xSPACE]"
930   goto660
940   play b$
950   goto660
960   char 1,1,10,"[40xSPACE]"
970   b$="":r=0
980   goto660
990   c=c+1
1000  vo$(c)=b$
1010  char 1,1,10,"[40xSPACE]"
1020  b$="":r=0:goto660
1030  for i=0toc
1040  play vo$(i)
1050  nexti
1060  goto660
1070  graphic 0:scnclr
1080  input "[SPACE]file[SPACE]naam[SPAC
E]?";fg$
1090  if fg$=""thengoto1080
1100  graphic 1,0
1110  dopen#2,(fg$),d0,u8,w
1120  print#2,c
1130  print#2,tp
1140  for i=1toc
1150  print#2,vo$(i)
1160  next i
1170  dclose#2
```

```
1180  goto660
1190  graphic 0:scnclr
1200  input "[SPACE]file[SPACE]naam[SPAC
      E]?";fg$
1210  if fg$=""thengoto1080
1220  graphic 1,0
1230  dopen#2,(fg$)
1240  input#2,c
1250  input#2,tp
1260  for i=1toc
1270  input#2,vo$(i)
1280  next i
1290  dclose#2
1300  tempo tp
1310  goto660
1320  graphic 0:scnclr
1330  input "[SPACE]tempo[SPACE]?[SPACE]
      ";tp
1340  if tp<1thentp=1
1350  if tp>255thentp=255
1360  tempo tp
1370  graphic 1,0
1380  goto660
1390  scnclr:fori=1to20:read z$:printz$:
      nexti
1400  data "[SPACE]gebruiks[SPACE]aanwij
      zing:"
1410  data "[SPACE]"
1420  data "[SPACE]f1[SPACE]-[SPACE]onth
      oud[SPACE]net[SPACE]geschreven[SPA
      CE]regel."
1430  data "[SPACE]f3[SPACE]-[SPACE]spee
      l[SPACE]alle[SPACE]regels[SPACE]af."
1440  data "[SPACE]f5[SPACE]-[SPACE]save
      [SPACE]muziekstuk[SPACE]op[SPACE]d
      isk."
1450  data "[SPACE]f7[SPACE]-[SPACE]laad
      [SPACE]muziekstuk[SPACE]van[SPACE]
      disk."
1460  data "[SPACE]"
1470  data "[SPACE]:[SPACE]-[SPACE]veran
      der[SPACE]muziek[SPACE]tempo"
1480  data "[SPACE]*[SPACE]-[SPACE]speel
      [SPACE]net[SPACE]geschreven[SPACE]
      regel."
1490  data "[SPACE][PIJL
LINKS][SPACE]-[SPACE]wis[S
      PACE]net[SPACE]geschreven[SPACE]re
      gel."
1500  data "[SPACE]~[SPACE]-[SPACE]wis[S
      PACE]heel[SPACE]muziekstuk."
1510  data "[SPACE]t[SPACE]-[SPACE]zet[S
      PACE]andere[SPACE]stem."
1520  data "[SPACE]o[SPACE]-[SPACE]zet[S
      PACE]1[SPACE]van[SPACE]de[SPACE]7[
      SPACE]oktaven.[SPACE](0-6)."
1530  data "[SPACE]u[SPACE]-[SPACE]volum
      e."
1540  data "[SPACE]v[SPACE]-[SPACE]zet[S
      PACE]1[SPACE]van[SPACE]de[SPACE]3[
      SPACE]stemmen."
1550  data "[SPACE]#(noot)[SPACE]-[SPACE
      ]de[SPACE]toon[SPACE]word[SPACE]ee
      n[SPACE]halve[SPACE]toon"
1560  data "[11xSPACE]hoger[SPACE]gespee
      ld."
1570  data "[SPACE]$(noot)[SPACE]-[SPACE
      ]de[SPACE]toon[SPACE]word[SPACE]ee
      n[SPACE]halve[SPACE]toon"
1580  data "[11xSPACE]lager[SPACE]gespee
      ld."
1590  data "[SPACE]w[SPACE]-[SPACE]de[SP
      ACE]volgende[SPACE]noten[SPACE]zij
      n[SPACE]hele[SPACE]noten"
1600  data "[SPACE]h[SPACE]-[SPACE]de[SP
      ACE]volgende[SPACE]noten[SPACE]zij
      n[SPACE]halve[SPACE]noten"
1610  data "[SPACE]q[SPACE]-[SPACE]de[SP
      ACE]volgende[SPACE]noten[SPACE]zij
      n[SPACE]1/4[SPACE]noten"
1620  data "[SPACE]i[SPACE]-[SPACE]de[SP
      ACE]volgende[SPACE]noten[SPACE]zij
      n[SPACE]1/8[SPACE]noten"
1630  data "[SPACE]s[SPACE]-[SPACE]de[SP
      ACE]volgende[SPACE]noten[SPACE]zij
      n[SPACE]1/16[SPACE]noten"
1640  data "[SPACE].[SPACE]-[SPACE]de[SP
      ACE]volgende[SPACE]noten[SPACE]wor
      den[SPACE]"
1650  data "[5xSPACE]de[SPACE]helft[SPAC
      E]van[SPACE]hun[SPACE]lengte[SPACE
      ]gespeeld"
1660  data "[SPACE]r[SPACE]-[SPACE]zet[S
      PACE]pause"
1670  data "[SPACE]/[SPACE]-[SPACE]einde
      [SPACE]programma"
1680  print:print"[SPACE]druk[SPACE]toet
      s":getkey z$
1690  scnclr:fori=1to8:readz$:printz$:ne
      xti
1700  print:print"[SPACE]druk[SPACE]toet
      s":getkey z$
1710  print:print:print"[SPACE]lijst[SPA
      CE]uit[SPACE]printen[SPACE][j/n]"
1720  get keyz$
1730  if z$="j"thengoto1760
1740  if z$="n"thengoto90
1750  goto1720
1760  restore
1770  open 4,4
1780  fori=1to28:readz$:print#4,z$:nexti
1790  close 4
1800  goto90
1810  fori=1toc
1820  vo$(i)=""
1830  nexti
1840  r=0:b$=""
1850  char 1,1,10,"[40xSPACE]"
1860  goto660
1870  graphic 0:scnclr:print"[2xSPACE]pr
      g.[SPACE]eindigen[SPACE][j/n][SPAC
      E]"
1880  getkeyhi$
1890  ifhi$="j"thengoto1910
1900  scnclr:graphic1,0:goto660
1910  key 1,"graphic"
1920  key 3,"directory"+chr$(13)
1930  key 5,"dsave"+chr$(34)
1940  key 7,"list"+chr$(13)
1950  scnclr:print:print"[3xSPACE]tot[SP
      ACE]de[SPACE]volgende[SPACE]keer..
      .[SPACE]"
1960  print"[18xCRSR-DOWN]":new
1970  resume
```

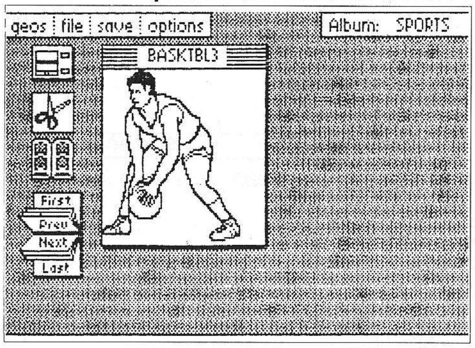
EINDE LISTING music maker

Checksum Music maker 128

regel 1	5	regel 330	37	regel 700	120	regel 1070	78
regel 2	76	regel 340	131	regel 710	57	regel 1080	74
regel 3	11	regel 350	137	regel 720	242	regel 1090	174
regel 4	173	regel 360	35	regel 730	175	regel 1100	116
regel 5	85	regel 370	76	regel 740	54	regel 1110	59
regel 10	134	regel 380	64	regel 750	142	regel 1120	144
regel 20	158	regel 390	133	regel 760	126	regel 1130	52
regel 30	111	regel 400	198	regel 770	120	regel 1140	238
regel 40	108	regel 410	71	regel 780	135	regel 1150	181
regel 50	190	regel 420	105	regel 790	141	regel 1160	121
regel 60	55	regel 430	180	regel 800	38	regel 1170	79
regel 70	72	regel 440	113	regel 810	254	regel 1180	164
regel 80	83	regel 450	152	regel 820	104	regel 1190	248
regel 90	238	regel 460	60	regel 830	246	regel 1200	120
regel 100	96	regel 470	54	regel 840	5	regel 1210	89
regel 110	196	regel 480	185	regel 850	236	regel 1220	33
regel 120	37	regel 490	47	regel 860	67	regel 1230	39
regel 130	77	regel 500	39	regel 870	74	regel 1240	57
regel 140	25	regel 510	139	regel 880	151	regel 1250	83
regel 150	239	regel 520	98	regel 890	206	regel 1260	142
regel 160	233	regel 530	152	regel 900	252	regel 1270	165
regel 170	238	regel 540	7	regel 910	13	regel 1280	224
regel 180	61	regel 550	39	regel 920	26	regel 1290	240
regel 190	81	regel 560	97	regel 930	185	regel 1300	79
regel 200	85	regel 570	192	regel 940	156	regel 1310	234
regel 210	43	regel 580	19	regel 950	127	regel 1320	102
regel 220	117	regel 590	207	regel 960	150	regel 1330	243
regel 230	30	regel 600	223	regel 970	20	regel 1340	138
regel 240	60	regel 610	31	regel 980	172	regel 1350	37
regel 250	109	regel 620	56	regel 990	39	regel 1360	180
regel 260	252	regel 630	177	regel 1000	162	regel 1370	80
regel 270	29	regel 640	190	regel 1010	160	regel 1380	169
regel 280	96	regel 650	133	regel 1020	195	regel 1390	255
regel 290	104	regel 660	235	regel 1030	204		
regel 300	104	regel 670	36	regel 1040	64		
regel 310	220	regel 680	83	regel 1050	172		
regel 320	129	regel 690	163	regel 1060	181		

Intergrated GEOS

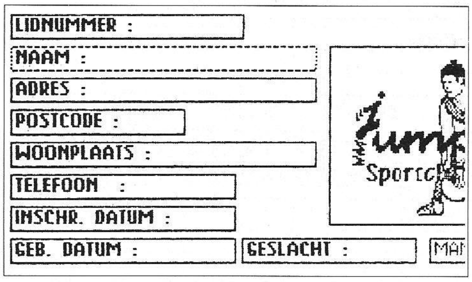
Het pakket GEOS is op zich al een fenomenaal programma, maar in combinatie met andere GEOS applicaties wordt de kracht van deze programmaserie pas echt duidelijk. Lees in dit artikel hoe u het geintegreerde karakter van deze software volledig kan benutten. In deze korte cursus GEOS integratie leert u de vele mogelijkheden van deze programma's toe te passen.



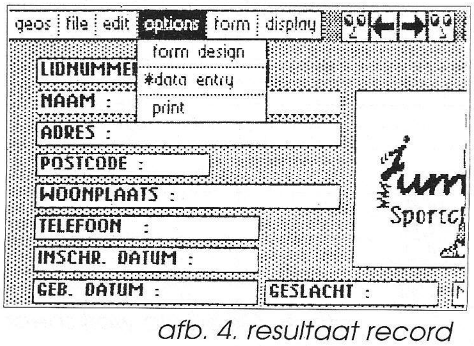
afb. 1. Graphics Grabber



afb. 2. GeoPaint bewerking



afb. 3. indeling record



afb. 4. resultaat record

In dit artikel wordt aandacht besteed aan het uitwisselen van gegevens uit de verschillende GEOS applicaties. De opzet van dit artikel heeft een educatief karakter. Hiermee wordt bedoeld dat u al werkend met de programma's leert wat de mogelijkheden van het totale pakket zijn. Op deze wijze ervaart u zelf hoe eenvoudig en handig de gehele GEOS serie op elkaar inspeelt. Daarnaast is het ontspannende bezigheid waar u menig uurtje aan bezig kunt zijn. Met name de textmanager en de photomanager zijn hierbij van groot belang. Zij vormen de "koppeling" van de gegevens in de onderlinge applicaties. Aan de hand van een gefingeerd praktijkvoorbeeld worden de verschillende mogelijkheden getoond.

Opzet en voorbereiding.

Het gegeven voor deze casus is een sportvereniging, met de naam JUMP, die een drietal sportaktiviteiten organiseert. Spelers kunnen zich als lid opgeven voor één van de aktiviteiten. Per maand is een contributie verschuldigd, waarmee de kosten per aktiviteit worden gedekt. De penningmeester van deze vereniging wil inzage hebben in de opbrengsten en uitgaven per aktiviteit. Voorts wil hij de leden volledige duidelijkheid verschaffen over de financiële verhandelingen van de vereniging. Hij heeft daarbij de beschikking over een Commodore 64 en de GEOS programmaserie. Tot zover de gegevens van de vereniging. Indien u de cursus volledig wilt volgen dient u over de volgende applicaties te beschikken. Allereerst GEOS 2.0 (een lagere versie voldoet ook), GeoFile, GeoCalc, DeskPack plus en GeoChart. Nu is het niet zo, als u één van deze applicaties niet in u bezit heeft (hetgeen wel jammer is) dat u niet mee kunt doen. U kunt dan gewoon de cursus volgen en slaat dat gedeelte over waar de ontbrekende applicatie voor nodig is. Voorts is enige kennis of ervaring betreffende de werking van de onderscheidenlijke programma's aan te bevelen. We zullen in de cursus niet te diep ingaan op de specifieke werking van elke applicatie op zich. Voor u begint is het raadzaam een nieuwe diskette te formatteren (in GEOS format) om de bestanden op weg te schrijven.

Het "grabben" van plaatjes.

De plaatjes die in het voorbeeld zijn ge bruikt komen van de Clip Art A en B diskette van NewsRoom, het zijn resp. BASKTBL3 en WORDS3. Deze plaatjes halen we op met behulp van de applicatie Graphics Grabber, zie afb. 1. Het is natuurlijk niet verplicht deze plaatjes te gebruiken, u mag zelf ook een ontwerp uitkiezen. En beschikt u niet over Graphics Grabber dan kunt u andere plaatjes uit PrintShop of PrintMaster halen met behulp van de PD programma's, die in de vorige Commodore Info nr. 6 zijn behandeld.

Plaats de beide plaatjes in een Photoalbum, die u "SPORTS" noemt. Copieer dit photoalbum naar een GeoPaint diskette en wijzig het ontwerp zoals in afb. 2 is weergegeven.

Let wel dat de tekst JUMP is ingelezen middels een photo scrap en de tekst SPORTCLUB is ingevoerd met behulp van de tekstmodus van GeoPaint. Tenslotte plaatst u het plaatje van de SPELER er in de tranparant modus overheen.

Kleur het totaal in met de invulpatronen van GeoPaint en plaats tenslotte een ka der om het geheel heen. Copieer het uiteindelijke resultaat naar een photo scrap en plaats het vervolgens in hetzelfde photoalbum "SPORTS".

Ontwerp database en invoer gegevens.

We gaan nu het GeoFile bestand aanmaken, zodat de essentiële gegevens van de vereniging ingevoerd kunnen worden. Copieer, voor zover dit nog niet is gebeurd het photoalbum naar uw werkdiskette. Bij het opstarten wordt een naam van het aan te maken bestand gevraagd. U noemt het bestand "LEDENBEST". Klik vervolgens de optie align on aan uit het display menu. U kunt de indicator uitschakelen, daar we deze toch niet nodig zullen hebben. Maak vervolgens het record aan volgens het voorbeeld zoals is weergegeven in afbeelding 3.

Eerst het plaatje invoegen en vervolgens de rubrieken. De rubriek NAAM wordt als sortfield gedefinieerd. Als het gehele record gereed is kunnen we beginnen met de invoer van gegevens, zie afb. 4.

Het is de bedoeling dat u minimaal 6 records invoert, zoals in het voorbeeld in afbeelding 5 is weergegeven. Dus 2 volleyballers, 2 basketballers en 2 handballers.

Laat hierbij ook de leeftijden variëren en het geslacht. Deze gegevens hebben we namelijk later weer nodig. Met behulp van dit bestand kunnen nu diverse lijsten worden vervaardigd. Er kan worden gesorteerd op bepaalde gegevens of met behulp van een zoekcriterium. In een aparte sub lay-out kan bijvoorbeeld een alfabetische lijst per sportaktiviteit worden afgedrukt.

Ook het maken van indexkaarten behoort tot de mogelijkheden. Op die manier kan een goed geordende kaartenbak worden opgezet. We dienen nu twee sub lay-outs te maken om verder te kunnen met de ingevoerde gegevens. De eerste sub lay-out dient naast het lidnummer, eveneens de naam, aktiviteit en de contributie te bevatten. Deze sub lay-out is bestemd voor een later te maken calc scrap. De tweede sub lay-out dient te bestaan uit lidnummer, naam, adres, postcode, woonplaats en geslacht, zie afb. 6.

Maak de keuze change lay out en zorg ervoor dat de sub lay-out voor de calc scrap op het scherm komt. Definieer nu een search form, zodat alle volleyballers worden geselecteerd. Maak hiervan een calc/text scrap met de optie build scrap uit het file menu. Copieer deze scrap naar een text album. Herhaal deze stappen voor de basketballers en de handballers, zie afb. 7.

Copieer dit text album naar de GeoCalc diskette. Vervolgens maakt u een merge data file, volgens de indeling zoals deze is weergegeven in afb. 6. Een merge data file maakt u door uit het file menu de optie build scrap te maken en vervolgens uit het verschijnende dialoogvenster de keuze merge data file.

Berekening van de contributie.

Maak een GeoCalc bestand onder de naam "BEREKENING". Maak de kolom D 22 karakters breed en de kolommen E t/m K ieder 7 karakters. Deel vervolgens de worksheet in volgens het voorbeeld van afbeelding 8. Nu kunt u met behulp van de text manager steeds gegevens ophalen van de respectievelijke leden per aktiviteit. Klik steeds de 1e cel in kolom A aan en middels de optie paste voegt u de betreffende gegevens in het respectievelijke blok in. Met behulp van de opties cut en paste kunt u de kopregel van elk blok steeds één regel omhoog verplaat sen afb. 9. Dit verhoogt de leesbaarheid. De maanden dient u zelf in te vullen. Alsmede de gegevens die in de betreffende kolommen moeten worden ingevuld. Nu gaan we de formules invoeren. Eerst klikken we cel K5 aan. We vullen nu als formule in =sum(E5:J5) en drukken op de ENTER toets. Met behulp van de copy optie zetten we dit gegeven in een calc scrap. Deze formule wordt nu met behulp

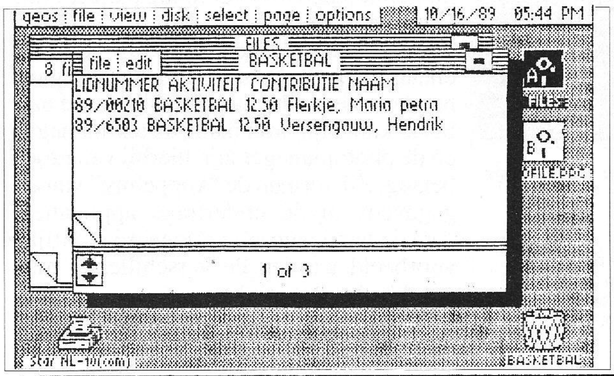
LIDNUMMER :	NAAM :	GEB. DATUM :	AKTIVITEIT :	AFDELING :	CON
89/6488	Flerkje, Jan Jaap	12-04-67	VOLLEYBAL	JUNIOREN	15.00
89/00210	Flerkje, Maria Petra	02-08-62	BASKETBAL	SENIOREN	12.50
89/00322	Flipsen, Kees	12-07-72	HANDBAL	JUNIOREN	10.00
89/8630	Gumbaljo, Victor Willem	23-05-78	HANDBAL	PUPILLEN	10.00
89/00321	Janssen, Johan	12-10-74	VOLLEYBAL	JUNIOREN	15.00
89/6503	Versengauw, Hendrik	10-02-56	BASKETBAL	SENIOREN	12.50

afb. 5. namenlijst

LIDNUMMER : | NAAM : | AKTIVITEIT : | CONTRIBUTIE p/m fl. :

LIDNUMMER : | NAAM : | ADRES : | POSTCODE : | WOONPLAATS :

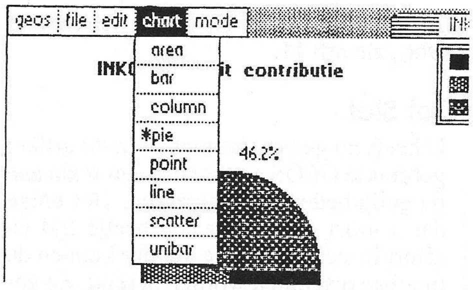
afb. 6. 2 sub lay-outs



afb. 7. calc/text scrap

	A	B	C	D	E	F	G	H	I	J	K
1	BEREKENING INKOMSTEN UIT CONTRIBUTIE										
2											
3	LIDNUMMER	AKTIVITEIT	CONTRIBUTIE	NAAM	JAN	FEB	MRT	APR	MEI	JUN	TOTAAL
4	----------	----------	----------	----------	------	------	------	------	------	------	------
5	89/00210	BASKETBAL	12.50	Flerkje, Maria petra	12.50	12.50	12.50	12.50	12.50	0.00	62.50
6	89/6503	BASKETBAL	12.50	Versengauw, Hendrik	12.50	12.50	12.50	0.00	0.00	0.00	37.50
7	----------	----------	----------	----------	------	------	------	------	------	------	------
8				TOTAAL							100.00
9											
10											
11	LIDNUMMER	AKTIVITEIT	CONTRIBUTIE	NAAM	JAN	FEB	MRT	APR	MEI	JUN	TOTAAL
12	----------	----------	----------	----------	------	------	------	------	------	------	------
13	89/00322	HANDBAL	10.00	Flipsen, Kees	10.00	10.00	10.00	10.00	10.00	0.00	50.00
14	89/8630	HANDBAL	10.00	Gumbaljo, Victor Willem	10.00	10.00	10.00	10.00	10.00	10.00	60.00
15	----------	----------	----------	----------	------	------	------	------	------	------	------
16				TOTAAL							110.00
17											
18											
19	LIDNUMMER	AKTIVITEIT	CONTRIBUTIE	NAAM	JAN	FEB	MRT	APR	MEI	JUN	TOTAAL
20	----------	----------	----------	----------	------	------	------	------	------	------	------
21	89/6488	VOLLEYBAL	15.00	Flerkje, Jan Jaap	15.00	15.00	15.00	15.00	15.00	15.00	90.00
22	89/00321	VOLLEYBAL	15.00	Janssen, Johan	15.00	15.00	15.00	15.00	15.00	15.00	90.00
23	----------	----------	----------	----------	------	------	------	------	------	------	------
24				TOTAAL							180.00
25											
26											
27	RECAPITULATIE INKOMSTEN en UITGAVEN										
28											
29		INKOMSTEN	UITGAVEN	SALDO							
30	----------	----------	----------	----------							
31	BASKETBAL	100.00	85.35	14.65							
32	HANDBAL	110.00	130.00	-20.00							
33	VOLLEYBAL	180.00	175.89	4.11							
34	----------	----------	----------	----------							
35	TOTAAL	390.00	391.24	-1.24							

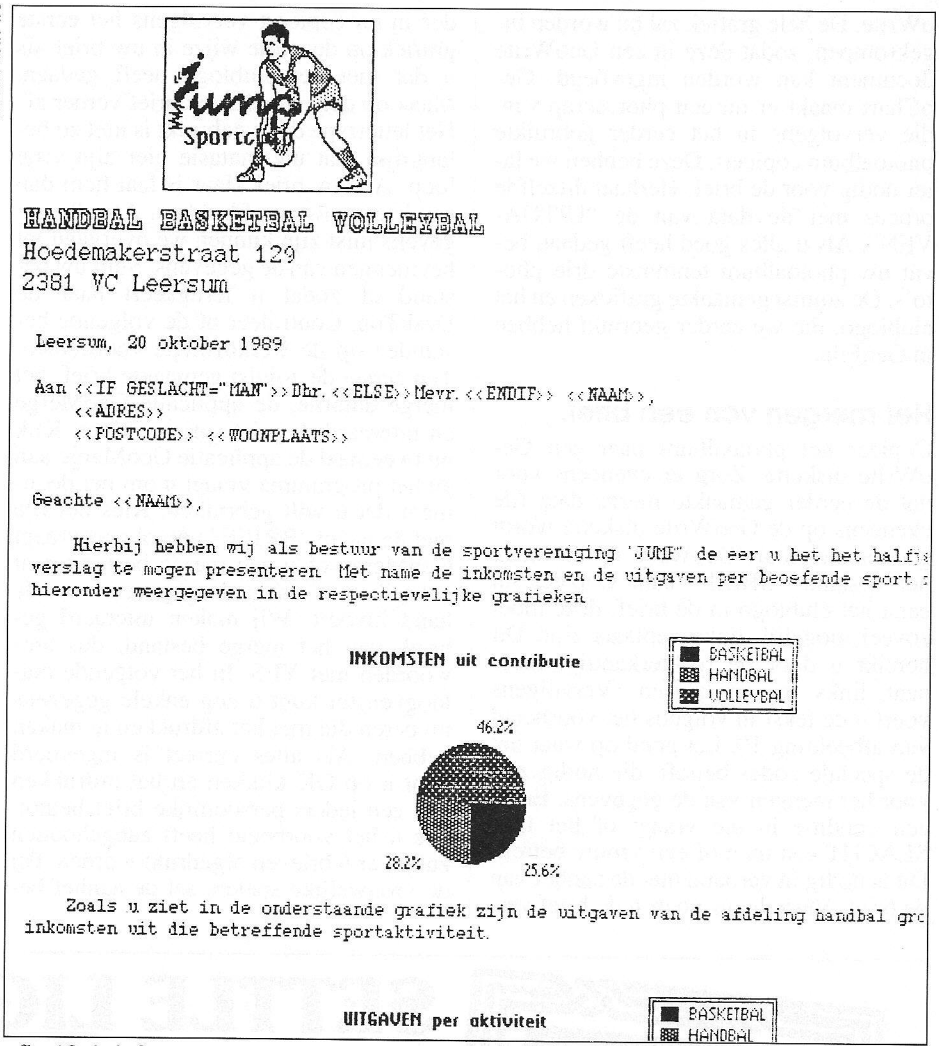
afb. 8. GeoCalc worksheet



van de optie paste ingevoegt in de volgende cellen K6, K13, K14, K21 en K22. Klik nu cel K8 aan en vul de volgende formule hier in =sum(K5:K6) gevolgd door RETURN. Copieer deze formule op de hiervoor beschreven wijze naar cel K16 en K24. Geef vervolgens de volgende cellen een naam middels de optie define name, K8 wordt BASKET, K16 wordt HAND en K24 wordt VOLLEY. We gaan nu de recapitulatie in- en uitgaven maken. Maak het blok A27 t/m D35 conform afbeelding 8. Voeg de gegevens voor de cellen B31 t/m B33 in met behulp van de optie paste name. Deze gegevens kunnen immers steeds wijzigen. Het voordeel hiervan is dat deze cellen dan steeds mee veranderen. De gegevens voor de cellen C31 t/m C33 vult u met hand in. De formule voor cel D31 luidt =C31-B31. Copieer deze formule naar de cellen D32 en D33, alsmede naar cel D35. Laat vervolgens de kolommen B31 t/m B33 en C31 t/m C33 optellen.

Overbrengen van gegevens.

Indien u dezelfde gegevens heeft gebruikt als in dit voorbeeld moet u tot de conclusie komen dat het saldo van de handballers negatief is komen te staan, enwel - fl. 20,00. Het totale saldo komt hiermee op - fl. 1,24. De penningmeester is dan voornemens een brief aan alle leden te sturen met daarin een explicatie van de financiële middelen. Hij is voornemens dit te doen middels een brief, aan ieder lid persoonlijk gericht met daarin een tweetal grafieken, die het een en ander verduidelijken. Deze brief zal zijn voorzien van het clublogo, zoals deze eerder is gebruikt in GeoFile. Eerst gaan we de grafieken maken met behulp van de gegevens die GeoCalc heeft opgeleverd. Daartoe moeten er een tweetal textscraps gemaakt worden, die in een textalbum worden geplaatst. De eerste textscrap moet de cellen A31 t/m B33 bevatten en de tweede loopt van C31 t/m C33. De eerste scrap renamen we in "INKOMSTEN" en de tweede noemen we "UITGAVEN". Copieer dit textalbum naar uw GeoChart diskette en start GeoChart op. Lees vervolgens de data van de scrap "INKOMSTEN" in en maak daarvan een taartdiagram. Plaats als koptitel "inkomsten uit contributie", zie afb. 9. Bewaar de grafiek middels de optie copy gevolgd door de optie to Ge-


Jump Sportclub

HANDBAL BASKETBAL VOLLEYBAL
Hoedemakerstraat 129
2381 VC Leersum

Leersum, 20 oktober 1989

Aan <<IF GESLACHT="MAN">>Dhr.<<ELSE>>Mevr.<<ENDIF>> <<NAAM>>,
<<ADRES>>
<<POSTCODE>> <<WOONPLAATS>>

Geachte <<NAAM>>,

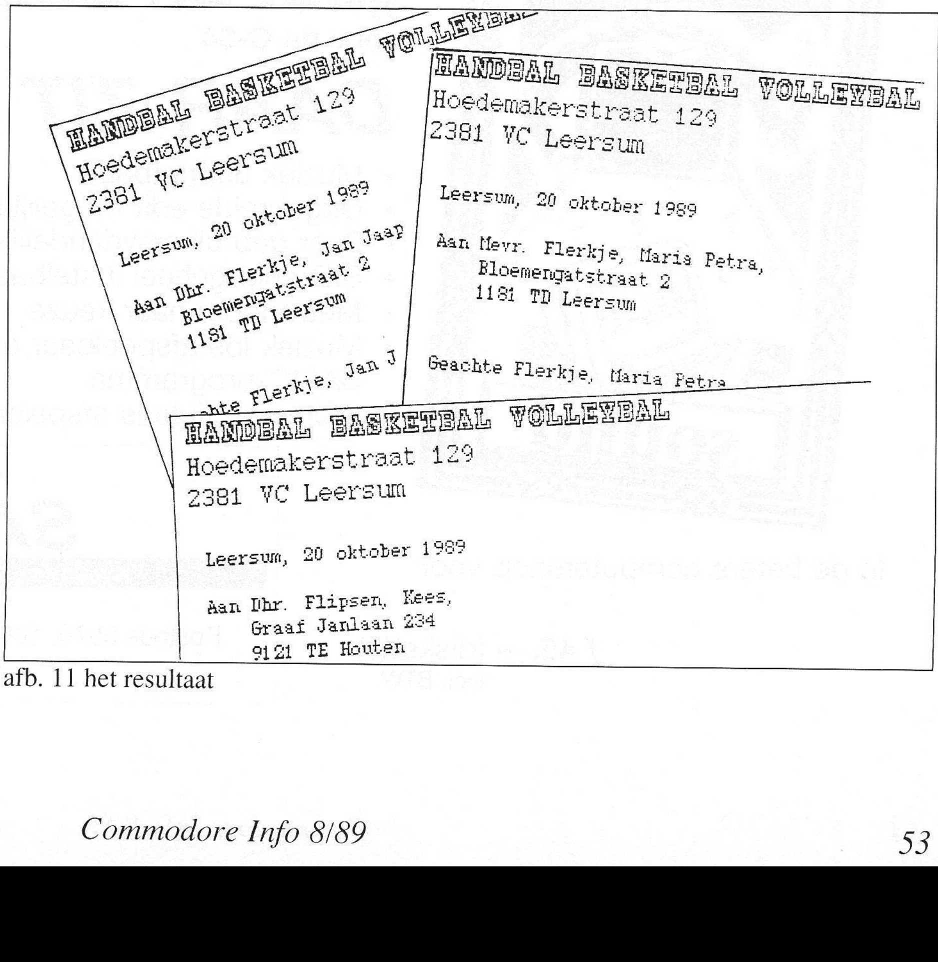
Hierbij hebben wij als bestuur van de sportvereniging 'JUMP' de eer u het het halfj verslag te mogen presenteren. Met name de inkomsten en de uitgaven per beoefende sport hieronder weergegeven in de respectievelijke grafieken.

INKOMSTEN uit contributie
BASKETBAL
HANDBAL
VOLLEYBAL
46.2%
28.2%
25.6%

Zoals u ziet in de onderstaande grafiek zijn de uitgaven van de afdeling handbal gr inkomsten uit die betreffende sportaktiviteit.

UITGAVEN per aktiviteit
BASKETBAL
HANDBAL

afb. 10. brief



afb. 11 het resultaat

oWrite. De hele grafiek zal nu worden ingekrompen, zodat deze in een GeoWrite document kan worden ingevoegd. GeoChart maakt er nu een photoscrap van, die vervolgens in het eerder gebruikte photoalbum copieert. Deze hebben we later nodig voor de brief. Herhaal ditzelfde proces met de data van de "UITGAVEN". Als u alles goed heeft gedaan bevat uw photoalbum tenminste drie photo's. De zojuist gemaakte grafieken en het clublogo, die we eerder gebruikt hebben in GeoFile.

Het mergen van een brief.

Copieer het photoalbum naar een GeoWrite diskette. Zorg er eveneens voor dat de eerder gemaakte merge data file eveneens op de GeoWrite diskette wordt gecopieerd. Start GeoWrite op en maak het bestand "BRIEF" aan. Copieer nu eerst het clublogo in de brief, deze moet zoveel mogelijk links geplaats zijn. Dit bereikt u door de rechterkantlijn meer naar links te verplaatsen. Vervolgens voert u de tekst in volgens het voorbeeld van afbeelding 10. Let goed op waar het de speciale codes betreft, die nodig zijn voor het mengen van de gegevens. Er zit een conditie in die vraagt of het GESLACHT een man of een vrouw betreft. Dit is nodig in verband met de aanhef van de brief. Vervolgens voert u de brief verder in en copieert vervolgens het eerste grafiek op dezelfde wijze in uw brief als u dat met het clublogo heeft gedaan. Maak op deze manier uw brief verder af. Het lettertype dat u gebruikt is niet zo belangrijk, laat uw fantasie hier zijn vrije loop. Als uw brief klaar is laat hem dan één keer gewoon afdrukken. Als alle gegevens juist zijn kunnen we overgaan tot het mengen van de gegevens. Sluit uw bestand af zodat u terugkeert naar de DeskTop. Controleer of de volgende bestanden op de werkdiskette voorkomen. Ten eerste de zojuist gemaakte brief, het merge datafile, de applicatie GeoMerge en uiteraard de juiste printerdriver. Klik nu tweemaal de applicatie GeoMerge aan en het programma vraagt u om het document dat u wilt gebruiken. Kies het file met de naam "BRIEF" vervolgens vraagt GeoMerge of u een merge bestand gaat gebruiken of dat u de gegevens met de hand invoert. Wij maken uiteraard gebruik van het merge bestand, dus antwoorden met YES. In het volgende dialoogvenster kunt u nog enkele gegevens invoeren die met het afdrukken te maken hebben. Als alles correct is ingevoerd kunt u op OK klikken en het afdrukken van een ieders persoonlijke brief begint. Als u het voorbeeld heeft aangehouden zullen er 6 brieven afgedrukt worden. Bij de vrouwelijke spelers zal de aanhef beginnen met Mevr. en bij de heren met Dhr., zie afb 11.

Tot Slot.

U heeft nu gezien dat met de in dit artikel gebruikte GEOS applicaties een scala aan mogelijkheden is te bereiken. Het enige dat u moet doen is er een beetje tijd en affort in steken. Op die manier kunnen de mooiste resultaten worden bereikt. Zo gebruik ik zelf alleen nog maar GEOS applicaties voor mijn huishoudadministratie, zoals een adressenbestand, het bijhouden van het maandverbruik aan gas en electriciteit. Met de andere applicaties, zoals GeoPublish zijn ook hele leuke dingen te doen. In combinatie met eerder gepubliceerde Public Domain programma's kunnen dan bv. schermen van spelletjes worden gegrabbed, die in GeoPaint verder bewerkt kunnen worden. Stuur uw leukste toepassing gemaakt met één of meerdere GEOS applicaties eens op. Misschien vinden we uw inzending wel zo leuk dat we hem plaatsen. Voor nu hoop ik dat u een leerzame les heeft gehad in het werken met GEOS.

B.V.

GEOS machinetaal cursus (7)

Bij wijze van sinterklaassurprise van de machinetaalcursus is een 3D labyrintspel voor GEOS gelist. Als rode draad -van Ariadne natuurlijk- hierdoorheen laat Peter Boncz de programmeertechnieken in een GEOS omgeving lopen. In deze aflevering aandacht voor onder andere de graphicsstring.

Om het tekort aan Public Domain software voor GEOS wat te lenigen, ben ik onlangs maar weer achter de C-64 gaan zitten. Hij stond er een beetje verlaten en stoffig bij, maar het uiteindelijke produkt, toen ik hem met vierkante ogen waarin een tollende blik scheen weer alleenliet, mag er volgens mij best zijn. Dit produkt is het Labyrint-spel, zoals ik hierboven al memoreerde. Het geheel is gehuisvest in maar liefst zes listings, die als het goed is allen geplaatst zijn. Het programma tekent in een venster steeds een (min of meer) driedimensionaal gezichtspunt in een doolhof, en er zijn pijlen, die met de muis aangeklikt kunnen worden, waarmee de speler zich kan bewegen. Dit alles is natuurlijk uitgevoerd in GEOS-huisstijl: trekmenu's, icons en dialog boxes (dialoogvensters). Het geliste programma is niet volledig, maar het werkt wel. Dit wil zeggen dat bij bepaalde routines, bijvoorbeeld die om een kaart van het labyrint op het scherm te brengen, nu als programmacode alleen een luttele RTS staat. De echte map-routine staat in de volgende aflevering van de GEOS machinetaalcursus. Iets soortgelijks geldt voor de plattegrond van het labyrint: er zit nu een plattegrondje van 10 kamers in, de volgende aflevering bevat een echte, uitdagende plattegrond van circa 220 kamers.

Include

Omdat labyrint een nogal uitgebreid programma is, heb ik in de listing gebruik gemaakt van het '.include' commando. Normaal wordt dit commando alleen gebruikt om aan het begin van de file de standaard GEOS bibliotheekfile geosSym en geosMac te importeren. Het is echter ook zelf door de programmeur te benutten om de listing wat overzichtelijker te maken. Ik heb bepaalde delen van het programma uit de hoofdlisting gehaald, en in aparte files gezet. Dit is gebeurd met alle teksten, de labyrint-kaart, en alle tabellen. Op de plaats in de hoofdlisting waar deze files met '.include filenaam' geimporteerd worden voegt de GeoAssembler ze dan in.

Menus

Illustratie 4 bevat de sublisting met alle tabellen. De tabel 'Menus' is de tabel voor het menu dat constant op het scherm staat. Het is een horizontaal menu, dit is te zien aan de derde regel, daar staat '.byte 2'. Op deze plaats in een menu-tabel staat altijd het aantal vakjes van het menu geORd met een 0 of een 128. Nul betekent horizontaal en 128 vertikaal. In een horizontaal menu overigens, verschijnen de vakjes van links naar rechts naast elkaar, in een vertikaal menu onder elkaar. Kortom er is dus een horizontaal menu met twee vakjes: GeosTxt ('geos') en LabyTxt ('labyrint'). Deze vakjes zijn allebei submenu's, dit is te zien aan '.byte 128'. In deze byte (na de tekstpointer) wijst een 0 altijd op een uitvoerroutine, en een 128 op een submenu. Als U zelf even verder kijkt, dan ziet U, dat er twee vertikale submenu's uit het hoofdmenu ontspruiten: het 'geos'-menu met de opties 'desk top' en 'quicktop', en het 'laby-

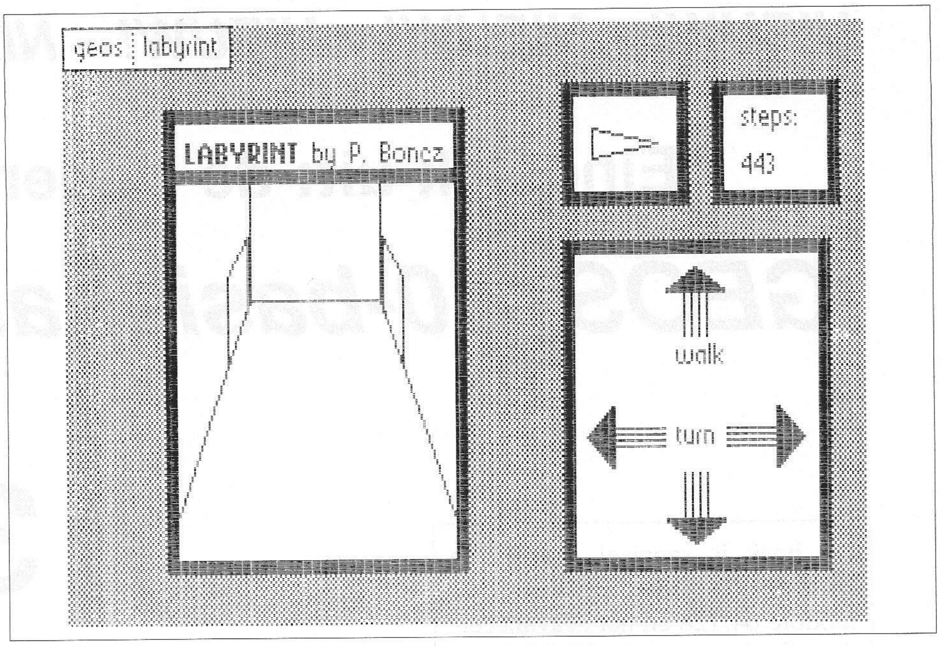


fig. 1 De HARDCOPY op thermisch papier

```
;kaartfile, moet op dezelfde
; disk als labyrintLst staan!
; deze kaartfile bevat een
; kleine kaart:
;
;      8     2
;
;      7  -  9
;
;     10  -  3  -  4
;
;      1  -  5  -  6
;
Kaart: .byte 10,5,0,0,1,10
       .byte 255,0,9,0,2,7
       .byte 0,4,0,10,2,9
       .byte 0,0,6,3,3,9
       .byte 0,6,0,1,2,10
       .byte 4,0,0,5,3,10
       .byte 8,9,10,0,1,8
       .byte 0,0,7,0,1,7
       .byte 2,0,0,7,2,8
       .byte 7,3,1,0,1,9
```

```
;tekstfile, moet op dezelfde
; disk als labyrintLst staan!
GeosTxt:        .byte "geos",0
DeskTopTxt:     .byte "desk top",0
QuickTxt:       .byte "quicktop",0
DESKTOPTXT:     .byte "DESK TOP",0
QUICKTOPTXT:    .byte "QUICKTOP",0
LabyTxt:        .byte "labyrint",0
MapTxt:         .byte "map",0
StartTxt:       .byte "start",0
InfoTxt:        .byte "info",0
```

fig. 2 Kaartfile-fig.3 Tekstfile

```
;tabelfile, moet op dezelfde disk als labyrintLst staan!
Menus:
  .byte 0,14
  .word 0,67
  .byte 2
  .word GeosTxt
  .byte 128
  .word GeosMenu
  .word LabyTxt
  .byte 128
  .word LabyMenu;

GeosMenu:
  .byte 14,41
  .word 0,47
  .byte 130
  .word DeskTopTxt
  .byte 0
  .word DeskTopRoutine
  .word QuickTxt
  .byte 0
  .word QuickRoutine;

LabyMenu:
  .byte 14,54
  .word 28,59
  .byte 131
  .word StartTxt
  .byte 0
  .word StartRoutine

  .word MapTxt
  .byte 0
  .word MapRoutine
  .word InfoTxt
  .byte 0
  .word InfoRoutine;

Icons:
  .byte 4
  .word 10
  .byte 10
  .word UpData
  .byte 30,81
  .byte 3,24
  .word UpRoutine
  .word DownData
  .byte 30,150
  .byte 3,25
  .word DownRoutine
  .word RightData
  .byte 33,128
  .byte 4,21
  .word RightRoutine
  .word LeftData
  .byte 26,128
  .byte 4,21
  .word LeftRoutine;
; bij het intikken nu eerst de rechterkolom, en dan wat hier
; onder staat.
UpData:     .byte 200,0,24,0,0,126,0,0,255,0,3,255,192,7,255,224
            .byte 31,255,248,63,255,252,255,255,255,255,255,255
            .byte 1,36,128,1,36,128,1,36,128,1,36,128,1,36,128
            .byte 1,36,128,1,36,128,1,36,128,1,36,128,1,36,128
            .byte 1,36,128,1,36,128,1,36,128,1,36,128,1,36,128
DownData:   .byte 200,1,36,128,1,36,128,1,36,128,1,36,128,1,36,128
            .byte 1,36,128,1,36,128,1,36,128,1,36,128,1,36,128
            .byte 1,36,128,1,36,128,1,36,128,1,36,128,1,36,128
            .byte 255,255,255,255,255,255,63,255,252,31,255,248
            .byte 7,255,224,3,255,192,0,255,0,0,126,0,0,24,0
RightData:  .byte 212,0,0,12,0,0,0,14,0,0,0,15,0,0,0,15,128,0,0,15
            .byte 192,0,0,15,224,0,0,15,240,255,255,255,248,0,0
            .byte 15,252,255,255,255,254,0,0,15,255,255,255,255
            .byte 254,0,0,15,252,255,255,255,248,0,0,15,240,0,0
            .byte 15,224,0,0,15,192,0,0,15,128,0,0,15,0,0,0,14
            .byte 0,0,0,12,0
LeftData:   .byte 212,0,48,0,0,0,112,0,0,0,240,0,0,1,240,0,0,3,2
            .byte 240,0,0,7,240,0,0,15,240,0,0,31,255,255,255,63
            .byte 240,0,0,127,255,255,255,255,240,0,0,127,255,255
            .byte 255,63,240,0,0,31,255,255,255,15,240,0,0,7,240,0,0
            .byte 3,240,0,0,1,240,0,0,0,240,0,0,0,112,0,0,0,48,0,0

GraphTable: .word up2nog,Nup2nog,righ2nog
            .word Nrigh2nog,lef2nog,Nlef2nog
            .word up1nog,Nup1nog,righ1nog
            .word Nrigh1nog,lef1nog,Nlef1nog
            .word upkeuze,Nupkeuze,righkeuze
            .word Nrighkeuze,lefkeuze,Nlefkeuze
up1nog:     .byte 1,45,0,163,2,90,0,53,1,155,0,163,2,110,0,53,0
Nup1nog:    .byte 1,45,0,163,2,74,0,91,2,126,0,91,2,155,0,163
            .byte 1,74,0,91,2,74,0,53,1,126,0,91,2,126,0,53,0
righ1nog:   .byte 1,127,0,95,2,127,0,70,2,135,0,83,2,135,0,114,0
Nrigh1nog:  .byte 0
lef1nog:    .byte 1,73,0,95,2,73,0,70,2,65,0,83,2,65,0,114,0
Nlef1nog:   .byte 0
up2nog:     .byte 1,45,0,163,2,90,0,53,1,155,0,163,2,110,0,53,0
Nup2nog:    .byte 1,45,0,163,2,87,0,60,2,113,0,60,2,155,0,163
            .byte 1,87,0,60,2,87,0,53,1,113,0,60,2,113,0,53,0
righ2nog:   .byte 1,115,0,63,2,115,0,53,2,120,0,65,2,120,0,76,0
Nrigh2nog:  .byte 0
lef2nog:    .byte 1,85,0,63,2,85,0,53,2,80,0,65,2,80,0,76,0
Nlef2nog:   .byte 0
upkeuze:    .byte 1,45,0,163,2,90,0,53,1,155,0,163,2,110,0,53,0
Nupkeuze:   .byte 1,45,0,163,2,60,0,127,2,140,0,127,2,155,0,163
            .byte 1,60,0,127,2,60,0,53,1,140,0,127,2,140,0,53,0
righkeuze:  .byte 1,145,0,138,2,145,0,88,2,155,0,113,0
Nrighkeuze: .byte 0
lefkeuze:   .byte 1,55,0,138,2,55,0,88,2,45,0,113,0
Nlefkeuze:  .byte 0
```

```
;                  labyrintHdr
.include     geosSym
.psect                             ; Ook voor de header nodig.
.word        0                     ; Begin van de header.
.byte        3                     ; Eerste 2 bytes altijd 0.
.byte        21                    ; Afmetingen Icon: 3 bytes
.byte        (63+$80)              ;  breed, 21 puntjes hoog.
.block 63                          ; 63 bytes icondata.
.byte        $83                   ; naar eigen smaak.
.byte        APPLICATION           ; Geos File Type USR.
.byte        SEQUENTIAL            ; APPLICATION = 6.
.word        ProgStart             ; SEQUENTIAL = 0.
.word        ProgEnd               ; laadadres.
.word        Labyrint              ; eindadres.
.byte        "Labyrint",0,0,0,0,0,0,0,0,0,0,0,0  ; startadres.
.byte        "Peter Boncz",0
.block 147
                                   ; aanvullen naar 255 bytes
```

```
                   labyrint.ink
;voor gebruik in de GeoLinker

.output      labyrint            ;naam van de uiteindelijke file
.header      labyrintHdr.rel     ;naam van de headerfile
.seq                             ;sequentiele filestructuur (niet VLIR)
.psect       $0900               ;begin programmakode
.ramsect     $4000               ;betekenisloos, want in labyrintLst is
                                 ; geen .ramsect gebruikt.
labyrintLst.rel                  ;naam van de programma-file
```

fig.4 Tabelfile-fig.5 LabyrintHdr-fig.6 labyrint.ink

```
   labyrintLst
;voor de GeoAssembler

.include geosSym
.include geosMac

;constanten
up        = 0
right     = 1
down      = 2
left      = 3
keuzefase = 2
nog1fase  = 1
nog2fase  = 0
WAAR      = 1
FOUT      = 0
maxsteps  = 16
mapviews  = 4
maxroom   = 10

.psect

ProgStart:
.include kaartfile
.include tekstfile
.include tabelfile
room:
   .byte 1
direction:
   .byte up
fase:
   .byte keuzefase
running:
   .byte FOUT
mapteller:
   .byte mapviews
steps:
   .byte 1,0

Labyrint:
  jsr ScreenInit
  LoadW r0,Icons
  jsr DoIcons
  lda #0
  LoadW r0,Menus
  jsr DoMenu
  LoadW $3a,10
  LoadB $3c,10
  rts;

StartRoutine:
  LoadB direction,up
  LoadB room,1
  LoadB fase,keuzefase
  LoadB running,WAAR
  LoadB mapteller,mapviews
  LoadW steps,maxsteps
  jsr Draw
  jmp InfoRoutine;

ScreenInit:
  LoadB dispBufferOn,192
  lda #2
  jsr SetPattern
  jsr i_Rectangle
  .byte 0,199
  .word 0,319
  lda #1
  jsr SetPattern
  jsr i_Rectangle
  .byte 28,182
  .word 40,160
  jsr i_Rectangle
  .byte 72,182
  .word 199,305
  jsr i_Rectangle
  .byte 20,60
  .word 199,249
  jsr i_Rectangle
  .byte 20,60
  .word 259,309
  lda #0
  jsr SetPattern
  jsr i_Rectangle
  .byte 33,47

  .word 45,155
  jsr i_Rectangle
  .byte 53,177
  .word 45,155
  jsr i_Rectangle
  .byte 77,177
  .word 204,300
  jsr i_Rectangle
  .byte 25,55
  .word 264,304
  jsr i_Rectangle
  .byte 25,55
  .word 204,244
```

listing labyrint

rint'-menu met de opties 'start', 'map' en 'info'.

Icons

In dezelfde illustratie (4) staat 'Icons',de icontabel. De eerste byte is een 4, wat erop duidt dat er vier icons zijn. Daarna staan resp. de X en Y coordinaat waar het muispijltje komt te staan na het op het scherm brengen van de icons. Daarna weer volgen de gegevens over de genoemde 4 icons. Eerst een pointer naar de data van het icon, daarna het aantal bytes van de x coordinaat (dus alleen x coordinaten 0,8,16,.. zijn mogelijk), dan de y coordinaat, dan de breedte van het icon in bytes, dan de hoogte in pixels, en dan tenslotte een uitvoerroutine voor als het icon aangeklikt wordt. De data van de iconplaatjes is in het GEOS- gecomprimeerde formaat opgesteld. Dit formaat is in een eerder stadium van mijn artikelen al bij GeoPaint aan de orde gekomen (de beruchte GeoPaint-catcher). Het is echt een te lang verhaal om dat nu weer helemaal uit te leggen. Kort en goed komt het er op neer dat de eerste byte een control-byte is, die aangeeft hoeveel losse bytes er volgen. Na deze eerste control-byte volgen van links naar rechts, en dan van onder naar beneden de bytes volgens het C-64 sprite systeem. De vier icons, waar ik het nu steeds over heb, zijn overigens 4 pijlen: naar boven, rechts, onder en links. Het resultaat is in illustratie 1 te zien, in het middelgrote vak rechtsonder staan onmiskenbaar de beschreven pijlen.

Plattegrond

Illustratie 2 bevat de kaartfile. De plattegrond die daarin staat heeft 10 kamers. Iedere '.byte' regel met daarachter 6 bytes stelt een kamer voor. De eerste regel bevat de gegevens van kamer 1, de tweede van kamer 2, en de 10-de van kamer 10. De eerste byte van zo'n regel is het kamernummer van de kamer boven, de tweede het kamernummer van de kamer rechts, de derde van de kamer onder, en de vierde van de kamer links. De laatste twee bytes op een regel zijn de coordinaten van de kamer in de plattegrond. Hierover in de volgende aflevering meer. Over de kamernummers nog dit: als een bepaalde kamer geen boven-, rechter-, onder- of linkerbuurman heeft dan wordt er een 0 neergezet. De waarde 255 heeft ook een speciale betekenis: dit is de uitgang van het doolhof. Verder zijn er nog twee andere vaste regels: kamer 1 is de kamer waarin de speler aan het begin van het spel wordt neergezet, en kamer 2 bevat de uitgang van het labyrint (de 255). Het hierboven beschreven 'labyrint-model' kan maximaal 254 kamers bevatten, want een byte heeft 256 mogelijke waarden, waarvan er twee (de 0 en 255) voor andere doeleinden beschikbaar zijn.

Suspense

Om de nodige 'suspense' (spanningsverhogende vertraging) aan te brengen kan er niet in een beweging van het ene kruispunt naar het andere gelopen worden. Dat moet met driemaal op een icon-pijl drukken. In deze driemaal klikken schuiven eventuele deuren steeds dichter naar de speler toe. Als men zich in een bepaalde kamer bevindt zijn er steeds drie mogenlijkheden:

1) er moeten nog twee stappen gedaan worden, en de deuren zijn nog ver weg. Dit heet in het programma de zg. 'nog2fase'.

2) er moet nog een stap gedaan worden, en de deuren zijn redelijk dichtbij. Dit is de zg. 'nog1fase'

3) Tenslotte is er de 'keuzefase', hier bevindt de speler zich op een 'kruispunt' en kan daadwerkelijk naar rechts of links afslaan.

Implicaties

Dit fasesysteem lijkt logisch, maar heeft de nodige implicaties. Als men in een bepaalde kamer is, in de nog1fase (redelijk dichtbij), en men draait 180 graden, dan moet nu dus de vorige kamer van ver te zien zijn (de nog2fase). Kortom bij 180 graden omdraaien (en niet bewegen) verandert men in de 'nog1fase' en de 'nog2fase' wel van kamer! Als er in de keuzefase, dus op een mogelijk kruispunt van verbindingen gedraaid wordt, dan blijft men wel in dezelfde kamer. Waar het dus allemaal op neer komt is dat de rechthoeken in de tekening van illustratie 2 overeenkomen met de keuzefases van de kamers, en de verbindingsstreepjes in tweeen gehakt dienen te worden. De dichtbije helft van een verbindingsstreepje correspondeert dan met de nog1fase en de verre helft met de nog2fase. Op deze wijze is ook makkelijk in te zien dat gegeven kamers A en B met een verbinding, de nog1fase van A de nog2fase van B is en de nog1fase van B de nog2fase van A.

Lopen

Voor het bewegen in het doolhof wordt het zogenaamde loopsysteem gehanteerd. Wij mensen lopen normaliter alleen naar voren. Willen we naar rechts, dan draaien we naar rechts, en lopen we weer naar voren. Zo werkt het ook in labyrint. De naar boven wijzende pijl zorgt ervoor dat er een stap naar voren wordt gedaan. Voor de engelstaligen onder U is deze pijl van het commentaar 'walk' voorzien. De pijlen naar rechts en links zorgen er respectievelijk voor dat er naar rechts en links gedraaid wordt. De pijl naar onderen tenslote, heeft een complete draaiing van het gezichtspunt (180 graden) tot gevolg.

Graphicsstring

In het plaatje van illustratie 1 is ook het venster met de 3D-blik in het labyrint te zien. Het tekenen van het plaatje wordt gedaan met de speciale GEOS-techniek van de zg. graphicsstring. Dat werkt zo: in r0 ($0002,$0003) staat een pointer naar een graphicsstring. Een graphicsstring is een serie bytes, afgesloten met een 0. De bytes 1,3,5,6 en 7 hebben een speciale betekenis en na deze bytes kunnen eventele databytes volgen. Waarom byte 4 niet in gebruik is, dat is weer zo'n levensvraag waar ik het antwoord echt niet op weet. Als er dus met de pointer naar de string in r0 een jsr GraphicsString gedaan wordt, dan gebeurt er het volgende al naar gelang van de inhoud van de string:

- .byte 1, en dan .word x .byte y: de 'pen' op (x,y) wordt gezet.
- .byte 2, en dan .word x .byte y: er wordt een lijn tussen de oude penpositie en de nieuwe, nu opgegeven, (x,y) getrokken, en de 'pen' wordt naar dit tweede punt verplaatst.
- .byte 3, en dan .word x .byte y: hetzelfde als hiervoor, alleen wordt er geen lijn, maar een opgevulde rechthoek getekend.
- .byte 5, en dan .byte pattern: hiermee wordt het lijnpatroon veranderd.
- .byte 6, en dan .byte string: de string wordt op de penpositie op het beeld gezet.
- .byte 7, en dan .word x .byte y: hiermee wordt een rechthoek getekend tussen de oude penpositie en de nieuwe (=x,y).

Let wel op, want het genoemde patroon waarmee de lijnen getekend worden is niet hetzelfde als het GEOS vulpatroon dat bijvoorbeeld bij i_Rectangle gebruikt wordt. Het 'lijn'patroon IS de lijn, d.w.z. bij een lijnpatroon van bijv. 1 worden er bij een horizontale lijn bytes van 1 naast elkaar gelegd. (d.w.z steeds 7 lege puntjes en 1 volle) Voor een normale lijn is dus patroon 255 (helemaal vol) nodig, dit is ook het patroon waarin standaard getekend wordt als er niets anders gedefinieerd is.

Layout

Terug naar de layout van het plaatje. In de tabelfile (illustratie 4) is de GraphTable gedefinieerd. Deze bestaat uit een opsomming van pointers naar graphicsstrings. Er zijn 18 pointers, voor 3 fasen: nog2fase, nog1fase en keuzefase. Bij iedere fase horen steeds 3 richtingen: alle zichtbare richtingen, namelijk up, right en left. Bij iedere richting tenslotte zijn er steeds 2 mogelijkheden: in die bepaalde richting wel of geen uitgang. En zoals U hoogstwaarschijnlijk al sinds de basisschool weet: 3*3*2=18. Nu correspondeert bijvoorbeeld 'up2nog' met de gra-

```
    jsr i_PutString
    .word 48
    .byte 45
    .byte 24,"LABYRINT",27
    .byte " by P. Boncz",0
    jsr i_PutString
    .word 243
    .byte 113
    .byte "walk",0
    jsr i_PutString
    .word 243
    .byte 140
    .byte "turn",0
    jsr i_PutString
    .word 270
    .byte 34
    .byte "steps:",0
    jsr Draw
    rts;

UpRoutine:
  lda running
  cmp #WAAR
  bne Uplbl3
  lda fase
  cmp #keuzefase
  bne Uplbl1
  ldy #up
  jsr DirectionInfo
  cpx #0
  beq Uplbl2
  cpx #2
  bne Uplbl0
  jmp Gewonnen
Uplbl0:
  sty room
  LoadB fase,#nog2fase
  jmp Uplbl2
Uplbl1:
  inc fase
Uplbl2:
  jsr Draw
Uplbl3:
  rts;

DownRoutine:
  lda running
  cmp #WAAR
  bne Downlbl2
  lda fase
  cmp #keuzefase
  beq Downlbl1
  ldy #down
  jsr DirectionInfo
  sty room
  lda fase
  cmp #nog1fase
  beq Downlbl0
  LoadB fase,#nog1fase
  jmp Downlbl1
Downlbl0:
  LoadB fase,#nog2fase
Downlbl1:
  inc direction
  inc direction
  jsr Draw
Downlbl2:
  rts;

RightRoutine:
  lda running
  cmp #WAAR


  beq Rightlbl0
  lda fase
  cmp #keuzefase
  beq Rightlbl0
  inc direction
  jsr Draw
Rightlbl0:
  rts;

LeftRoutine:
  lda running
  cmp #WAAR
  beq Leftlbl0
  lda fase
  cmp #keuzefase
  beq Leftlbl0
  inc direction
  inc direction
  inc direction
  jsr Draw
Leftlbl0:
  rts;

Draw:
  lda #1
  jsr SetPattern
  jsr i_Rectangle
  .byte 53,177
  .word 45,155
  LoadB dispBufferOn,64
  lda #0
  jsr SetPattern
  jsr i_Rectangle
  .byte 53,177
  .word 45,155
  lda #1
  jsr SetPattern
  LoadW a0,GraphTable
  LoadB a1H,0
  LoadB a2L,up
Drawlbl0:
  lda fase
  clc
  adc fase
  adc fase
  sta a1L
  asl a1L
  asl a1L
  lda a1H
  adc a1H
  adc a1H
  adc a1H
  adc a1L
  sta a1L
  ldy a2L
  jsr DirectionInfo
  cpx #0
  bne Drawlbl1
  inc a1L
  inc a1L
Drawlbl1:
  ldy a1L
  lda (a0),y
  sta r0L
  iny
  lda (a0),y
  sta r0H
  jsr GraphicsString
  inc a1H
  inc a2L
  lda a2L
  cmp #down
  bne Drawlbl2
  lda #left
  sta a2L
Drawlbl2:
  lda a1H
  cmp #3
  bne Drawlbl0
  LoadB dispBufferOn,192
  jsr i_RecoverRectangle
  .byte 53,177
  .word 45,155
  dec steps
  lda steps
  cmp #255
  bne Drawlbl3
  dec steps+1
  lda steps+1
  cmp #255
  bne Drawlbl3
  jmp Verloren
Drawlbl3:
  jsr DrawSteps
  jsr DrawCompass
  rts;

DirectionInfo:
  sty a4L
  lda direction
  clc
  adc a4L
  and #3
  tay
  lda #0
  sta a4H
  lda room
  sta a6L
  dec a6L
  clc
  lda a6L
  adc a6L
  sta a4L
  lda #0
  adc a4H
  sta a4H
  clc
  lda a6L
  adc a4L
  sta a4L
  lda #0
  adc a4H
  sta a4H
  asl a4L
  rol a4H
  LoadW a5,Kaart
  clc
  lda a4L
  adc a5L
  sta a5L
  lda a4H
  adc a5H
  sta a5H
  lda (a5),y
  tay
  cmp #0
  beq Dirlbl0
  cmp #255
  beq Dirlbl1
  ldx #1
  jmp Dirlbl2
Dirlbl0:
  ldx #0
  jmp Dirlbl2
Dirlbl1:
  ldx #2
Dirlbl2:
  rts;

Gewonnen:
Verloren:
  LoadB running,#FOUT
DrawSteps:
DrawCompass:
  rts;
DesktopRoutine:
QuickRoutine:
  jsr EnterDesktop
InfoRoutine:
MapRoutine:
  jsr GotoFirstMenu
  rts;

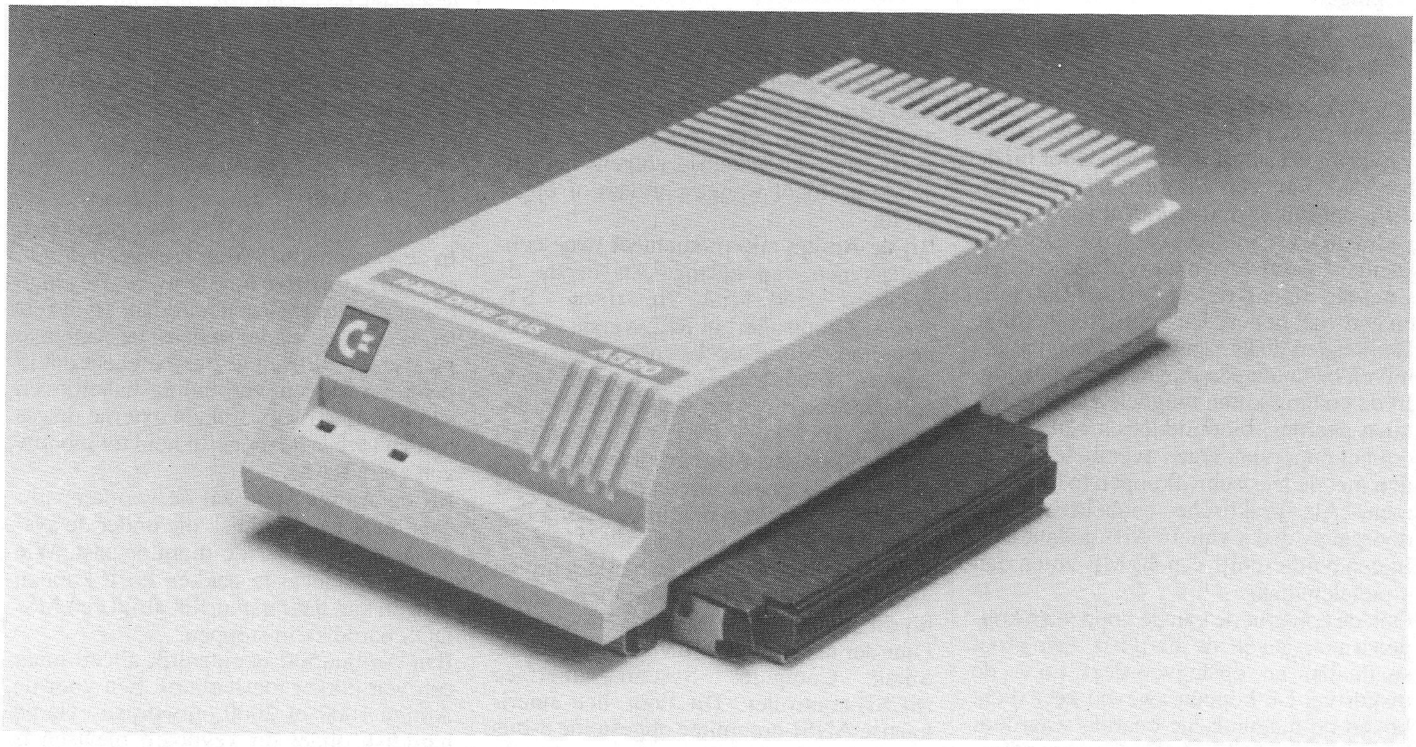
ProgEnd:
```

phicsstring voor een kamer in de nog2fase met een uitgang naar voren, en 'Nrighkeuze' met de graphicsstring voor een kamer in de keuzefase zonder een uitgang naar rechts. Om een complete kamer te tekenen, dienen er altijd 3 strings gegraphicsstring-d te worden: een voor up, een voor right en een voor left. De kamer die in illustratie 1 te zien is, werd dus verkregen door achtereenvolgens een Nup1nog, een righ1nog, en een lef1nog te graphicsstring-en. Dit tekenen gebeurt in de routine 'Draw'.

In de volgende aflevering zal ik nog wat meer uitleg geven bij 'Draw', en een andere belangrijke routine: 'DirectionInfo'. Voorzover de listing nu in het blad staat, kan de speler starten, en een beetje door het doolhof lopen. De 'quicktop', 'info' en 'map' opties uit het hoofdmenu werken nog niet. Ook het kompas en het 'steps' venster (te zien in illustratie 1) staan nog buiten werking. Om het geval nu aan de praat te krijgen moeten allereerst natuurlijk al de files met Geowrite ingetikt worden. De files 'labyrintLst' en 'labyrintHdr' dienen met de GeoAssembler gecompileerd te worden. Tenslotte moet de file 'labyrint.ink' door de Geolinker gehaald worden. De uiteindelijke, werkende, file heet dan 'labyrint.ex'. Volgende keer ben ik terug met de rest van het programma!

Peter Boncz

Amiga harddisks



De voordelen van een harde schijf zijn wijd en zijd bekend. Een grote opslagcapaciteit en een aanmerkelijk snellere data-accesstijd dan bij een floppystation maken de harddisk onmisbaar voor de power-user. Zo ook bij de Amiga waar grafische pracht en rekenkracht om een harde schijf roepen.

Veel voordelen tegen een stevige prijs
Data- en programmabestanden worden steeds groter. Iedereen die zich het begin van het Commodore PET en C-64 tijdperk nog herinnert weet dat vroeger meerdere files op één cassettebandje pastten. Na enkele minuten laden kon je dan een aardig spelletje spelen of mooie brieven schrijven. Die tijden zijn nu vrijwel voorbij. Een beetje Amiga-programma heeft al één of twee 3.5 inch diskettes nodig en dat geeft praktische laad- en opslagproblemen. Bovendien staat de gebruiker voortdurend lang te wachten op het inladen of wegschrijven. Zelfs als er twee diskdrives gebruikt worden blijft het allemaal traag en veel schijfjes wisselen. Een harddisk ondervangt dit probleem. Daar passen met gemak vele tientallen diskettes aan informatie op en zijn programma- en datafiles binnen één handbereik. Het tweede grote voordeel van de harde schijf is dat de laad- en wegschrijftijden aanmerkelijk korter zijn dan bij de diskdrive het geval is.
Ondanks de geboden voordelen beschikt nog lang niet elke Amiga- gebruiker over een machine met harddisk. Dit in tegenstelling tot de IBM-compatibele PC waar een harde schijf eerder regel dan uitzondering is. Daarvoor zijn twee belangrijke redenen aan te voeren:
Amiga-harddisks zijn aanmerkelijk duurder dan hun gelijkwaardige PC-collega's

- Behalve bij de Amiga 2000-serie is er niet in een interne diskdrive voorzien

Het aspect "duurder" vormt voor menige krap gebudgetteerde hobbyist(e) een probleem. Terwijl bij de IBM PC-kloon er al een 20 MB voor *f* 600,- of een 40 MB voor *f* 950,- model in de systeemkast gaat betaal je bij de Amiga voor een 20 MB harddisk al minimaal *f* 1.200,-. En dat is lang niet mis. Waar komt dit prijsverschil vandaan? Een veel gehoord argument is het relatief kleine aantal Amiga's in de wereld. Kleinere aantallen Amiga- drives maken de stukprijs duurder. De eerlijkheid gebied ons echter op te merken dat de Amiga-harddisk in Nederland in vergelijking met bijvoorbeeld Duitsland behoorlijk duur zijn.
Een tweede argument betreft het feit dat de Amiga 1000 en 500 intern geen plaats hebben voor een harde schijf. Dat betekent een extra kastje, interface en een eigen voeding met bijkomende meerprijs. Bij de Amiga 2000-familie werd daarentegen wel rekening gehouden met een harddiskcontroller en inbouwruimte voor een extra drive-unit of hardcard (= harddisk op insteekkaart).

Wat is een harddisk?

In feite is een harde schijf niets anders dan een soort super- diskdrive. Het gaat allemaal groter en sneller. Het principe blijft echter hetzelfde.
Een standaard Amiga 3.5 inch 80 tracks diskette is een flexibel magneetschijfje (de disk) binnen een stevige plastic hoes. Op dit flexibele schijfje passen 880 KiloByte (onder MS-DOS slechts 720 KB) aan data. De hoeveelheid data die per tijdseenheid geschreven kunnen worden hangt af van:

- De draaisnelheid van het schijfje. Hoe harder hoe meer data er per tijdseenheid overgedragen kunnen worden. In de praktijk ligt de rotatiesnelheid van de harddisk tussen de 3.000 en 4.000 toeren per minuut
- De efficiëncy van de koppenbewegingen
- De aard van de besturingselectronica, de drive- controller

Een flexibel schijfje kan, tenzij speciaal geconstrueerd, niet snel draaien want dan vervormt het schijfoppervlak. Door bij de harddisk van één of meerdere metalen disks gebruik te maken kan er veel sneller gedraaid worden. Die veel hogere rotatiesnelheid geeft een data-opslagcapaciteit van vele MegaBytes (1 MB = 1.000 KB) en een veel hogere leesschrijfsnelheid.
De metalen disks zijn veelal van aluminium en bevatten een magnetische coating. In de coating zitten magnetiseerbare metalen deeltjes, bindmiddel en een harder om het oppervlak tegen eventuele botsingen met de lees/schrijfkoppen te beschermen. Als praktische eenheid worden doorgaans disks van 10 MB gehanteerd. In een harde schijf van 40 MB zitten dan 4 metalen disks.
Ook de constructie van de koppen en koppenarmen geven de harddisk een groot snelheids- en opslagvoordeel t.o.v. de diskdrive. De koppen zweven zeer dicht boven de magnetische coating waardoor het magnetische lees/schrijf-proces effectiever (in een hogere sector- en spoordichtheid) verloopt. Speciale snelle electromotoren van het voice-coil- of stappentype zorgen voor een effectieve beweging van de koppen van track naar track. Deze techniek gaat heel wat verder dan de rudimentaire armbewegingen in de diskdrive. De koppen zelf zijn aerodynamisch gevormde magneetspoeltjes die zeer dicht boven het coatingsoppervlak jakkeren. Een door de vorm van de koppen gecreeerd luchtkussen beschermt tegen een crash in de coating.
De aandrijfmotor van de disks is een hoogwaardig precisie- instrument. Zelfsmerend, duurzaam en zeer geringe slijtage gevoeligheid garanderen een gemiddeld leven van circa vijf bedrijfsjaren.
De controller zorgt voor de besturing van de harddisk en de protocollering van het datatransport. Bij de Amiga 500 en 1000 is er sprake van een speciale Amiga-controller. Dit interface zit doorgaans in het drive-huis gebouwd. Bij de Amiga 2000-serie is er keuze uit speciale Amiga-harddisks en via het bridgeboard tevens uit de veel goedkopere PC-drives. In het laatste geval krijgt het Amiga-deel een eigen partitie op de DOS-harddisk toebedeeld.

Controller-typen

Tot de taken van het harddisk-besturingsinterface behoren:

- Besturingsbevelen aan de on board electronica met electromotoren van de harddisk door te geven
- Het datatransport tussen de CPU, het RAM en de harddisk te regelen

In de praktijk ontvangt de controller besturingsbevelen van de Amiga soft- en hardware en vertaalt deze naar de harddisk. De tijdens dit proces genereerde datastromen worden keurig volgens het geldende protocol weggeschreven of opgehaald.
Bij de Amiga zijn momenteel twee controllertypen van belang. Als eerste de Standard Industrial Interface ST-506/412-controller, in RLL-versie OMTI genoemd. Momenteel een enigzins verouderd, tamelijk traag (er is geen eigen spooradressering!) en weinig flexibel interface. Toch kent deze (van oorsprong) Seagate-standaard een groot toepassingsgebied en is goedkoop aan te maken. U vindt dit type met name in de voordelige harddiskklasse. Op één ST- 506/412-controller kunnen slechts twee harde schijven worden aangesloten. De foutherkenning bij dit controllertype laat te wensen over.
Duurder maar sneller en veelzijdiger is de Small Computer System Interface (SCSI)-controller. Dit door het amerikaanse ANSI-commitee opgestelde 8-bits controller-protocol is moderner als zijn Seagate-collega. Het gaat om een semi-intelligent en vooral logisch computerinterface met capaciteit voor zeven devices. Die devices kunnen zowel harddisks, printers als diskdrives zijn. Gezien de verscheidenheid aan devices is het noodzakelijk dat elk apparaat zijn eigen controller met microprocessor en wat RAM aan boord heeft om optimaal met de computer te kunnen communiceren. Dat maakt het systeem zowel universeler als duurder.
Alle SCSI-controllers maken gebruik van een vastgelegde commandoreeks, de Common Command Set, een een 8-bits databus. De commandoset verzorgt de standaardfuncties zoals formattering, het gebruik van buffers, foutherkenning en correctie.

Wel/geen DMA

Direct Memory Access is een gevleugeld woord bij de omschrijving van de systeemsnelheidsprestaties. DMA betekent niets meer of minder dat er met zo min mogelijk bemoeienis van de Motorola 680X0 datablokken door de controller worden verplaats. Alle CPU- tijd die daarbij uitbespaart wordt betekent pure winst op de systeemtijd en maakt de Amiga dus sneller.
Indien de controller de data rechtstreeks van de harddisk naar het RAM en visa versa kan verplaatsen wordt er veel op processor I/O- tijd bespaart. Bij de specificaties van diverse harddiskmodellen is het echter oppassen geblazen. Vaak is het helemaal niet duidelijk of er nu van een volledige DMA-transfer tussen de SCSI-drive in het systeem-RAM of slechts van een cache-RAM sprake is. In het laatste geval krijgt u mindere resultaten dan bij een echt DMA-systeem.

In- of extern?

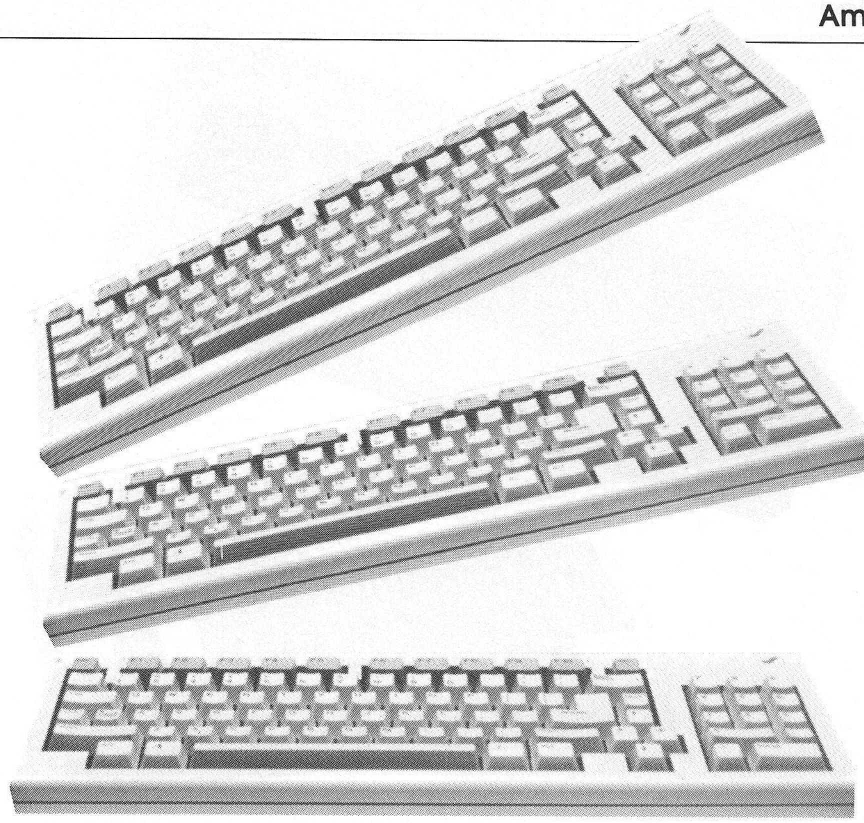
Bij de Amiga 500 en 1000 is er geen keus. In de systeemkast is domweg geen ruimte om een extra drive in te bouwen. De enige manier om er toch een harddisk bij te zetten is om deze onder of naast de Amiga te plaatsen. Praktisch gezien betekent dat in ieder geval een verbindingskabeltje en een netsnoer meer. Enkele externe drives bieden tevens de mogelijkheid tot inbouw van extra RAM.
Bij de Amiga 1000 zal de voorkeur uitgaan naar een harddisk die onder de systeemkast of onder de monitor past. Wie nog een Sidecar te pakken heeft kunnen krijgen kan daar natuurlijk altijd een MS-DOS harddisk in stoppen.
Een Amiga 500 is eigenlijk alleen maar een wat dikker toetsenbord. Een voor de Amiga 1000 of 2000 ontworpen externe harddisk onder dit keyboard plaatsen is veelal geen gezicht of werkt onhandig. Over blijven een positie naast de Amiga 500 (daarvoor zijn inmiddels diverse instijl harddisks verkrijgbaar) en een plat model onder de monitor. Een alternatief zijn de systeem ombouwkasten waarin meerdere drives passen en die samen met de 500-toetsenbordkast één attractief geheel vormen.
Bij de Amiga 2000-serie is er keuze uit een in- of externe harddisk. In het geval van een intern-model kunt u nog kiezen uit een losse drive-unit met controllor of een hardcard in een uitbreidingsslot. Een externe drive is duurder en maakt het totale systeem er wat lijviger op. De echte power-user die meerdere harddisks via een SCSI-interface meerdere harddisks wil aansturen kan overwegen om een drive-uitbreidingsmodule aan te schaffen. Een intern-alternatief kan de Golem Amiga-tower zijn waarin u vijf drives kwijt kunt.

MS-DOS

Zoals reeds eerder werd opgemerkt zijn harddisks voor IBM- compatibele PC's

Lees verder op pagina 69

Inzake invoer



Zonder invoer zou er geen data zijn. Zonder data zou een programma nutteloos zijn. Zonder een programma zou een computer nutteloos zijn. Zonder een computer zou een computerblad nutteloos zijn. Zie hier het nut van invoer.

Zoals de bovenstaande kop al doet vermoeden, we gaan het over invoer hebben. Beter gezegd, we gaan het over invoer vanaf het toetsenbord hebben.

Wanneer u al enigzins bekend bent in C, dan zult u weten dat op de Amiga op een aantal verschillende manieren invoer vanaf het toetsenbord kan verwerken. Even een kleine opsomming, de eerste, invoer met behulp van de functie scanf(). Dit is niet bepaald de ideale functie daar het een aantal gebreken heeft. Bij het inlezen van getallen is de functie scanf() wel zeer gemakkelijk. Als tweede kennen we uit de standaard C bibliotheek, de functies getc() en getchar(). U leest hiermee één karakter vanaf het toetsenbord. Er is wel een nadeel. Wilt u, zoals met éénvoudige programma's vaak is, invoer vanuit het DOS venster krijgen dan zult u een RETURN moeten geven. Dé facto betekent dit dus dat u niet één karakter leest maar twee. Het ingetypte karakter en een RETURN. Het is tevens zo dat de functies getc() en getchar() de invoer bufferen zodat u in principe veel meer karakters in kunt voeren. Op zich geen probleem zult u stellen. Toch wel. Denk bijvoorbeeld aan een invoer routine die met vaste grenzen zit waarover niet geschreven mag worden. Het programma kan dit niet controleren dus gebeurt dit wel.

Een derde mogelijkheid. Invoer rechtstreeks van het keyboard.device of input.device. Dit is de meest doeltreffende en tegelijk de meest complexe manier. Een fout is hierbij zo maar gemaakt. Pak een goed boek over de devices van de Amiga, ga een maand zitten lezen, en ga vervolgens een maand experimenteren. Dan zult u de techniek onder de knie hebben. De vierde mogelijkheid is door gebruik te maken van de Gadgets. Het is mogelijk een string gadget te openen en hiermee stringinvoer te verkrijgen. Dit is absoluut de meest nette manier, de absolute 'Amiga way' en tegelijk de meest programmeur geestdodende oplossing. Perfect voor snelle invoer en zonder meer goed ogend op het beeldscherm. Toch willen we het onszelf ook weer niet zo gemakkelijk maken. De vijfde en tevens laatste oplossing. Lees de invoer met behulp van de Message Ports uit het actuele window. Deze laatste methode gaan we in dit artikel volgen, vanwege de niet al te moeilijke manier om karakters in te lezen en toch houden we voldoende eigen programmatische (geen bestaand Nederlands, we weten het!) controle over de invoerroutine.

Invoer uit een window

Elk window beschikt over een faciliteit die IDCMP genoemd wordt. Deze afkorting staat voor Intuition Direct Communication Port. Een IDCMP verwerkt en filtert gegevens van een aantal devices waaronder het input.device die op zijn beurt gegevens krijgt vanuit het keyboard.device. Door nu een bepaalde IDCMP flag te zetten kan men kiezen tussen 'rauwe' toetsenbord (=RAWKEY) invoer of ASCII (=VANILLAKEY) invoer. Wat doet RAWKEY? Deze levert, zoals we al gezegd hebben, de ruwe, onbewerkte toetsenbord codes aan. Dit zijn dus nog geen ASCII codes. Het voordeel van RAWKEY is, dat u van alle toetsen een andere code terugkrijgt. Lees de toetscode uit het element 'Code' van de structure IntuiMessage. Om te controleren of één van de shift-, control- of alttoetsen is ingedrukt dient u het element 'Qualifier' te controleren. Deze code dient u daarna nog om te zetten naar een ASCII code.

Gemakkelijker is de optie VANILLAKEY. Deze levert rechtstreekse ASCII codes aan. Er zijn hier echter wel een aantal nadelen aan verbonden. De cursor toetsen worden niet uitgelezen, de functietoetsen worden overgeslagen en de toetsen van het numerieke toetsenbord zijn niet verschillend van hun tegenhangers van het alfanumerieke toetsenbord. De opgeleverde ASCII codes zijn bij VANILLAKEY afhankelijk van het met setmap gekozen toetsenbord configuratie.

Hoe kies je nu de instelling RAWKEY of VANILLAKEY. Ook hier zijn weer twee manieren voor. Ten eerste kun je de optie meegeven als je het window zelf opent. Vul RAWKEY of VANILLAKEY in het element 'IDCMPFlags' van de NewWindow structuur in, en de mogelijkheid is daar.

Als het window al geopend is dan is er een tweede manier om de IDCMP instellingen te veranderen. Gebruik de functie ModifyIDCMP(). ModifyIDCMP() kent twee parameters;

ModifyIDCMP(pointer naar een window,de nieuwe flags). Wilt u bijvoorbeeld alleen de mogelijkheid VANILLAKEY toevoegen, dan zult u de originele flags van het window moeten OR'en met de nieuwe flags. Een voorbeeld;

```
flags=MijnWindow->IDCMPFlags;
ModifyIDCMP(MijnWindow,flags|VANILLAKEY);
```



Het window zal nu klaar staan om gegevens in te lezen vanaf het toetsenbord. Het geeft hierbij ASCII tekens terug.

Hoe de toetsen te lezen?

Zoals we al eerder vermeld hebben komen de toetsengegevens terecht in een structure genaamd 'IntuiMessage'. Dit gebeurt echter niet vanzelf. U dient hiervoor wel een functie aan te roepen die de IntuiMessage structuur opvult met de data. Deze functie heet GetMsg(). U geeft als parameter de pointer naar de UserPort van het huidige window mee op. Bijvoorbeeld;

```
TestMelding=(struct IntuiMessage *)
GetMsg(TestWindow->UserPort);
```

Hierbij komen de gegevens terecht in de variabele TestMelding welke van het type 'struct IntuiMessage' is. Er bestaat echter wel een voorwaarde, de melding moet weer geretourneerd worden met behulp van de functie ReplyMsg(). Als parameter gaat hierbij mee, de pointer naar de IntuiMessage structuur. Even terug naar het bovenstaande voorbeeld. Het voorbeeld zou dan herschreven moeten worden naar;

```
TestMelding=(struct IntuiMessage *)
GetMsg(TestWindow->UserPort);
ReplyMsg(TestMelding);
```

Zou u de melding niet retourneren dan volgt onvermijdelijk een Guru-meditatie. Dit is nog niet voldoende. Als er geen toetsinvoer zou zijn, zou TestMelding geen data bevatten en zou dus leeg zijn.

We moeten dus wachten tot er een toets ingedrukt is. Met behulp van de functie Wait() kunnen we de task, waarin het programma draait, laten wachten totdat een signaal optreed. Dit signaal is in dit voorbeeld afkomstig van het toetsenbord. Het voorbeeld veranderd nu weer enigzins;

```
Wait(1<<TestWindow->UserPort->mp_Sig
Bit);
Test Melding=(struct IntuiMessage *)
ReplyMsg(TestMelding);
Class=TestMelding->Class;
Code=TestMelding->Code;
```

De variabele Class vermeld nu uit wat voor, van te voren opgegeven, IDCMP bron de melding afkomstig is. Stel, u heeft ook nog gadget invoer, welke u laat detecteren met de optie GADGETUP. In het element Class zal GADGETUP vermeld staan als u een Gadget geselecteerd heeft. Er is wel één uitzondering. Bij de Aztek compiler. Wilt u toetseninvoer verkrijgen met behulp van de optie VANILLAKEY dan zult u moeten testen op NULL. Class bevat dan namelijk de waarde nul. (=NULL)

Het programma

Het programma is uit een aantal functies en procedures opgebouwd, te weten GetKey(), PrintC(), NewInput(), Quit() en NewStrLen(). Wat doet welke functie?
GetKey() leest op de hierboven besproken manier het toetsenbord uit.
PrintC() print, zoals u al vermoed, het teken op het beeldscherm. Het genereert hierbij een cursor en houd rekening met RETURN en BACKSPACE.

NewInput() is het eigenlijke hoofdprogramma. GetKey() en PrintC() kunnen niet zondermeer in een willekeurig programma gebruikt worden. Hiervoor dient u aan enkele voorwaarden te voldoen. U moet gebruik maken van de globale variabelen IntuitionBase en GfxBase. Deze dienen al bekend te zijn, en de libraries dienen al geopend te zijn, voordat u gebruik maakt van GetKey() en PrintC(). NewInput() opent deze libraries als het nodig is, reserveert geheugenruimte en roept de diverse ondergeschikte functies aan.
Quit() is een zeer simpele beeindigingsroutine. Het print een boodschap af en geeft een waarde voor DOS af. Als er een fout optreed dan zal dit 5 (=WARN) zijn. NewStrLen() is een extraatje daar de normale functies om de stringlengte te bepalen niet gebruikt kunnen worden. Het berekent de lengte van een in een geheugendeel geplaatste string.

Parameters van NewInput()

NewInput() vereist 6 parameters. Hoe ziet NewInput() eruit?

```
NewInput(Opslag,x,y,Length,Color,Tekst,
BackGround)
```

De eerste parameter genaamd Opslag. Hier dient u een adres naar een door uzelf gecreëerde bufferlocatie in te vullen. Bijvoorbeeld,

```
UBYTE *MijnBuffer;
```

NewInput(MijnBuffer, ...

U mag ook het geheugen reserveren met AllocMem(), het principe blijft ook in dat geval ongewijzigd.
De tweede en derde parameter zijn x en y locatie. Houd hierbij wel in gedachte dat het hier gaat om karakterlocaties. De x waarden liggen bij lage resolutie tussen 0 en 39 en bij hoge resolutie tussen 0 en 79. De y waarden liggen bij non-interlace tussen 0 en 24. Bij overscan kunt u er nog 2 bij op tellen. Bij interlace liggen de waarden tussen 0 en 48.
De volgende. Parameter nummer vier. Hier moet u het aantal tekens, dat u in wilt voeren, invullen. Het resultaat op het scherm zal dan zijn dat er, afhankelijk van het hier ingevulde getal, een aantal BackGround tekens afgedrukt worden.
De vijfde parameter, Color, moet een integer waarde bevatten welke de kleur aangeeft waarin u de uitvoer geschreven wilt hebben. In DOS heeft u alleen de keuze uit 0,1,2 en 3. De één na laatste parameter, Tekst, moet een pointer naar een startstring bevatten. Wilt u geen string dan dient u hier NULL in te vullen. Deze startstring zal voor de invoerregel geprint worden.

De laatste parameter. Van oorsprong hadden we de routine zo geschreven dat deze alleen punten kon afprinten als u de BACKSPACE toets indrukte. Nu kunt u elk karakter wat u wilt zien hier specificeren inclusief een SPATIE.
NewInput() heeft geen globale variabelen of vreemde functies nodig die u zelf opmoet geven. Dit in tegenstelling tot GetKey() en PrintC(). U kunt NewInput() dus vanuit elke willekeurige routine oproepen. GetKey() verondersteld dat de 'intuition.library' en de 'graphics.library' geopend zijn. Dit geldt ook voor PrintC().

Parameters van NewStrLen(d)

Dit is een zeer korte functie. Het heeft één parameter nodig, te weten de pointer naar de buffer waar de tekststring opgeslagen is. Het geeft dan een integer waarde terug welke de lengte van de string aangeeft. Onthoud wel, DE AFSLUITENDE NUL WORDT NIET MEEGEREKEND!

NewInput() in libraryvorm

De nu volgende opmerkingen zijn alleen van nut voor Aztek C bezitters. Lattice C bezitters dienen een main() body aan het programma te breiden. Voor Aztek C bezitters dus. Het is mogelijk het programma newinput.c in een library te zetten zodat u alleen de library bij het linken op hoeft te geven. Dit lijkt lastig, doch het is zeer eenvoudig.
Compile eerst het programma, cc newinput.c +L
Als de compiler en de assembler geweest zijn wacht dan even met linken. Roep nu de LIBRARIAN aan op de volgende wijze.

lb new.lib newinput.o

De functies zullen nu opgeslagen zijn in de library genaamd 'new.lib'
Typ vervolgens de listing 'demo.c' in. Compile deze met,

cc demo.c +L
en link deze met,
ln demo.o -lnew -lc32

Nu is het uit te voeren door het commando 'demo' in te typen. En voila, c'est une miracle!
Zoals u ziet wordt nu de library 'new.lib' meegelinkt. In deze library is onder andere de functie NewInput() opgeslagen.
Het programma is voorzien van begeleidend commentaar zodat het merendeel voor zichzelf spreekt. That's all for now, let the listing (vrij omgezet) begin! Eek, het zit er weer op. Waar blijft de tijd, en de pegels?

JOHAN & JOHAN.

Instructies bij NEWINPUT.C

```
Programmanaam        : newinput.c
Programmeertaal      : C
Gebruikt programma : Aztek C v3.4 en hoger
Opties :
                cc newinput.c +L
                lb new.lib newinput.o
Uitvoer              : GEEN.
```

```
Programma 1.    NEWINPUT.C

#include "functions.h"
#include "exec/types.h"
#include "exec/memory.h"
#include "graphics/gfxbase.h"
#include "graphics/gfxmacros.h"
#include "intuition/intuitionbase.h"

#define RP CurScreen->RastPort

extern struct IntuitionBase *IntuitionBase;
extern struct GfxBase *GfxBase;

USHORT GetKey(CurWindow)
struct Window *CurWindow;
{
 USHORT Class,Code;
 register struct IntuiMessage *m1;
 struct IntuiMessage *IMes;

 Class=Code=NULL;
 Wait(1L << CurWindow->UserPort->mp_SigBit);
                                          /*    wacht op toetsdruk
    */
 IMes=(struct IntuiMessage *)GetMsg(CurWindow->UserPort);
                                          /* haal gegevens op uit
    IDCMP */
 ReplyMsg(IMes);                        /* geef de message weer door
    */
 Class=IMes->Class;                     /* Class = soort message
    */
 Code=IMes->Code;                       /* Code  = de toetsencode
    */

 if((m1=(struct IntuiMessage
    *)GetMsg(CurWindow->UserPort))!=NULL)
    {
     while(m1-Qualifier==IEQUALIFIER_REPEAT && m1 != NULL)
       {
        ReplyMsg(m1);
        m1=(struct IntuiMessage *)GetMsg(CurWindow->UserPort);
   }
    }                                     /* als er evt. herhalings-
    toetsen zijn, lees deze
                                                dan uit.
    */
 if(Class==NULL)
    return(Code);                       /* Als Class = VANILLAKEY
    */
 else
    return(NULL);                       /* anders een NULL
    teruggeven */
}

void PrintC(CurScreen,Key_x,Key_y,chr,TmpA,BackGround)
struct Screen *CurScreen;
SHORT Key_x,Key_y;
char chr;
UBYTE *TmpA,BackGround;
{
                                          /* genereer een cursor met
```

```
                                       behulp
   van de Blitter          */
 BltBitMap(&CurScreen->BitMap,Key_x,Key_y-6,
          &CurScreen->BitMap,Key_x,Key_y-6,
          8,8,0x30,0x01,TmpA);
 switch(chr)
   {
    case(0x08):                   /* als chr
   = BACKSPACE           */
      Move(&RP,Key_x,Key_y);
      Text(&RP,&BackGround,1);

   BltBitMap(&CurScreen->BitMap,Key_x,Key_y-6,

   &CurScreen->BitMap,Key_x,Key_y-6,
             16,8,0x30,0x01,TmpA);
      break;

    case(0x0d):                   /* als chr
   = RETURN              */
      break;

    default:                      /* voer
   anders de tekst uit
                                       en
   verplaats de cursor      */
      Move(&RP,Key_x,Key_y);
      Text(&RP,&chr,1);

   BltBitMap(&CurScreen->BitMap,Key_x+8,Key_y-6
   ,

   &CurScreen->BitMap,Key_x+8,Key_y-6,
             8,8,0x30,0x01,TmpA);
      break;
   }
}

void
   NewInput(Opslag,x,y,Length,Color,Tekst,BackG
   round)
UBYTE *Opslag[80],Tekst[],BackGround;
SHORT x,y,Length,Color;
{
 register int i;
 struct Screen *CurScreen;
 struct Window *CurWindow;
 ULONG Flags;
 SHORT BaseX,BaseY,c,Teller;
 UBYTE *TmpA;
 BOOL IntFlag,GfxFlag;

IntFlag=GfxFlag=FALSE;  /* controleer of
   IntuitionBase
            en GfxBase al geopend zijn */
 if(IntuitionBase==NULL)        /* zo niet,
   open ze dan          */
  {
   if(!(IntuitionBase=(struct IntuitionBase *)
      OpenLibrary("intuition.library",0L)))
        Quit(5,"Geen intuition.library\n");
   IntFlag=TRUE;
  }           /* gebruik IntFlag en GfxFlag
          om de libraries aan het
          eind van de sobroutine
          weer te sluiten           */
 if(GfxBase==NULL)
  {
   if(!(GfxBase=(struct GfxBase *)
      OpenLibrary("graphics.library",0L)))
        Quit(5,"Geen graphics.library\n");
   GfxFlag=TRUE;
  }
              /* reserveer geheugen voor
          Blitter operaties          */
if(!(TmpA=(UBYTE *)

   AllocMem(RASSIZE(640,256),MEMF_CLEAR|MEMF_CH
   IP)))
    Quit(5,"Geen geheugen voor TmpA\n");
           /* wis buffer voor teken-
        opslag en reset variabelen */
Teller=BaseX=BaseY=c=NULL;

for(i=0;i<80;i++)
  Opslag[i]=NULL;
             /* haal pointers naar het
       actuele scherm en venster  */
CurScreen=IntuitionBase->ActiveScreen;
CurWindow=IntuitionBase->ActiveWindow;
flags=CurWindow->Flags;        /* haal IDCMP
   Flags....       */

ModifyIDCMP(CurWindow,flags|VANILLAKEY);
           /* ..en verander deze          */
SetDrMd(&RP,JAM2);
SetBPen(&RP,0);
SetAPen(&RP,1);                     /* set
   tekenmodes             */

x*=8;
y*=8;

Move(&RP,x,y);
if(Tekst!=NULL)
  Text(&RP,Tekst,strlen(Tekst)); /* print
   startstring       */

x+=(8*strlen(Tekst));
BaseX=x;                         /* zet
   uitgangspositie
        achter de startstring         */
for(i=0;i<Length;i++)      /* print
   achtergrondtekens       */

   PrintC(CurScreen,(x+=8),y,BackGround,TmpA,Ba
   ckGround);

            /* genereer cursor              */
BltBitMap(&CurScreen->BitMap,x+8,y-6,
          &CurScreen->BitMap,x+8,y-6,
          8,8,0x30,0x01,TmpA);

x=BaseX;
BltBitMap(&CurScreen->BitMap,x+8,y-6,
          &CurScreen->BitMap,x+8,y-6,
          8,8,0x30,0x01,TmpA);

do {
   c=GetKey(CurWindow);           /* haal
   karaktercodes op        */
   switch(c)
    {
     case(0x08):                     /* if c ==
   BACKSPACE            */
       if(Teller>0)
         {
          Opslag[Teller]=(UBYTE *)NULL;

   PrintC(CurScreen,x,y,(char)c,TmpA,BackGround
   );

          x-=8;
          Teller-;
         }
       break;

     default: /* en de defaultuitvoer is..     */
       if(Teller<Length)
         {
          Opslag[Teller]=(UBYTE *)c;

   PrintC(CurScreen,(x+=8),y,(char)c,TmpA,BackG
```

```
   round);
         Teller++;
       }
     break;
   }
 }while(c!=0x0d);                  /* totdat de
   RETURN ingedrukt
             wordt                                  */
 ModifyIDCMP(CurWindow,flags);/* restore
   flags naar oude
             waarden                                */
 FreeMem(TmpA,RASSIZE(640,256));
 if(GfxFlag)
   CloseLibrary(GfxBase);
 if(IntFlag)
   CloseLibrary(IntuitionBase);
}

Quit(how,why)
int how;
char *why;
{
 printf("%s",why);
 exit(how);
}

int NewStrLen(Opslag)
UBYTE *Opslag[200];
{
 register int i;

 i=0;
 while(Opslag[++i]!=NULL);       /* verhoog i
   totdat er een
                 NULL aangetroffen wordt */
 return(i-1);/* wilt u het NULL teken wel

   meegeteld hebben, zet
                     hier dan neer:
                  return(i);                 */
}
```

```
Instructies bij DEMO.C
----------------------
Programmanaam       : demo.c
Programmeertaal     : C
Gebruikt programma  : Aztek C v3.4 en hoger
Opties :
        cc demo.c +L
        ln demo.o -lc32
Uitvoer             : demo

Programma 2   DEMO.C
-------------------------------- 7------
#include "functions.h"
#include "exec/types.h"

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;

main()
{
 UBYTE *Opslag[80];
 int i;

 NewInput(Opslag,3,10,30,1,"Uw antwoord:",'.');
 printf("\n\nLengte
   ->%2d\n",NewStrLen(Opslag));
 printf("Opgeslagen is ->");

 for(i=0;i<NewStrLen(Opslag);i++)
   printf("%c",Opslag[i]);
 printf("\n\n");
}
```

Vervolg van pagina 62

goedkoper dan die voor de Amiga. Alleen bij de Amiga 2000-familie bestaat de mogelijkheid om van een MS-DOS harddisk gebruik te maken. Daarvoor is dan wel een bridgeboard noodzakelijk en dat maakt het er ook weer niet goedkoper op.

Gaat u zowel onder DOS als het Amiga-OS werken kies dan voor een harddisk van 40 MB of meer. Zowel MS-DOS als het Amiga-OS gebruiken namelijk een eigen harddiskdeel (dat zich als een afzonderlijke harddisk gedraagt) en dat verminderd de beschikbare opslagcapaciteit. 20 MB per Operating Systeem is tegenwoordig wel het minste dat u nodig heeft.

De harddisk vergroot de mogelijkheden en kracht van de Amiga aanzienlijk. Even door de zure financiële appel heenbijten en u krijgt er veel gebruikersgemak en efficiëncy voor terug. Menige kersverse harddiskbezitter heeft al verzucht: "Had ik het maar veel eerder gedaan".

U.S.

Harddisk	MB	AMIGA	Contrl.	In/extern	Autoboot	DMA	Prijs-indicatie	fabrikant
Combitec HD 40	40	500	OMTI5520	Extern	Ja	Nee	ƒ 1.600	Combitec
Supradrive	30	500	SCSI	Extern	Ja	Nee	ƒ 2.000	Supra Corp.
- als hardcard	30	2000	SCSI	Intern	Ja	Nee	ƒ 2.000	Supra Corp.
Impact A500	20	500	SCSI	Extern	Ja	Nee	ƒ 1.700	GVP
System 2000	20	500/1000	OMTI	Extern	Ja	Nee	ƒ 1.100	Vortex
Amigos-Boil	40	500/1000	OMTI	Extern	Ja	Nee	ƒ 1.600	Amigos
X-tension	40	1000	OMTI	Extern	Ja	Nee	ƒ 2.200	Seagate
Double Disk	20	500	OMTI	Extern	Ja	Nee	ƒ 1.000	Reis-ware
Hard-Disk A590	20	500	SCSI	Extern	Ja	Nee	ƒ 1.300	Commodore
Message AHD-20	20	500/1000	OMTI	Extern	Ja	Nee	ƒ 1.500	Message/Seagate
Amiga Hardd-disk	20	500/1000	OMTI	Extern	Ja	Ja	ƒ 1.100	Alcomp
Kupke HD3000	40	500/1000	OMTI	Extern	Ja	Nee	ƒ 1.100	Kupke/Golem
A 2090A + ST506	40	2000	SCSI	Intern	Ja	beperkt	ƒ 2.000	Commodore
Impact A500	20	500	SCSI	Extern	Ja	Nee	ƒ 1.700	GVP
Impact A2000	40	2000	SCSI	Intern	Ja	Cache	ƒ 2.400	GVP
Hardframe	40	2000	SCSI	Intern	Ja	Ja	ƒ 2.400	Microbotics
Superperformance	30	500/1000	OMTI	Extern	Ja	Nee	ƒ 1.250	Weisgerber
- als Filecard	30	2000	SCSI	Intern	Ja	Nee	ƒ 1.200	Weisgerber
Flashcard + 20MB	20	2000	SCSI	Intern	Ja	Cache	ƒ 1.200	Expansion Techn.
TrumpCard 20 MB	20	2000	SCSI	Intern	Ja	Nee	ƒ 1.100	Interactive Video
Trumpcard 500	20	500	SCSI	Extern	Ja	Nee	ƒ 1.400	Interactive Video
Escort	32	500/1000	SCSI	Extern	Ja	Nee	ƒ 2.000	Expansion Techn.
SA 500	40	500	SCSI	Extern	Ja	Nee	ƒ 3.500	Comspec
Fasttrak Jr	20	500	SCSI	Extern	Ja	Nee	ƒ 1.400	Xetec
Fasttrak	40	500	SCSI	Extern	Ja	Ja	ƒ 2.500	Xetec
SA500/Bernoulli	20	500	SCSI	Extern	Ja	Nee	ƒ 4.000	Comspec/Iomega
Skyline CHD-AG	20	500/1000	OMTI	Extern	Ja	Nee	ƒ 950	Skyline
Golem FileCard	40	2000	OMTI	Intern	Ja	Nee	ƒ 1.350	Golem

Desktopvideo op de Amiga

Video als complete thuisstudio



Desktopvideo is het samen met de computer van achter het bureau perfecte videoproducties maken. De Amiga functioneert daarbij als titel- of logogenerator, animator, special effect generator (SEG) of grafisch artiest. Via een kleine investering in hard- en software breidt u de Amiga 500, 1000 of 2000 uit tot een complete video- of kabelkrant-studio. Doe daar nog eens de kwaliteit van het moderne VHS-systeem bij en de geboden mogelijkheden reiken net zo ver als de eigen fantasie.Desktopvideo op de Amiga-computer en video als complete thuisstudio

Vroeger was de thuis en low- tot no-budget video een tamelijk statisch gebeuren. De amateur en kleine vakman kon niet meer dan de opgenomen videoscenes netjes achter elkaar te monteren, het geluid mixen en met een titelgenerator wat simpele titeltjes maken. Wie special effects, wipes, paint-effecten, mozaïek beelden, scrollende veelkleurige titels, logotransformaties, animaties en superpositie wou kon diep in de beurs tasten. Een beetje professionele videostudio kost zo'n vijf jaar geleden meer dan *f* 100.000,-.
Toen kwam de Amiga 1000 op de markt. Van huis uit was deze Motorola MC 68000 PC voorbereid voor magnifieke videopresentaties. Helaas alleen in RGB en voor Europa nog niet in het PAL composiet videosysteem voor opname op homerecorders en (semi-) professionele VHS-machines. Toch werd video-potentieel van de Amiga al snel door onafhankelijke hardwareleveranciers en tal van knutselende pioniers ontgonnen. De eerste *genlocks* boden niet zo'n beste beeldkwaliteit, maar het werd wel ineens mogelijk om Amiga beelden met bestaande of video-opnamen van Kamera's te kombineren. Dit naar keuze in de voorgrond (bijvoorbeeld superpositie van titelletters) of in de achtergrond (live video in computer-scenery). Ook verschenen er kwalitatief goede *PAL- generatoren* om grafische beelden uit de Amiga rechtstreeks naar de video of kabel (bij kabelkranten) over te spelen.
Circa twee jaar na de introductie van de Amiga 1000 kwamen de eerste professionele genlocks voor de Amiga-familie op de markt. De prijzen waren eerst zeer hoog. Nu kost een kwaliteits genlock gemiddeld iets meer dan *f* 2.000,- en een goed amateurmodel circa *f* 1.000,-. Ook Commodore zelf kwam met een inbouw genlockkaart en speciale PAL-videokaart voor de Amiga 2000.
Momenteel is er een behoorlijke keuze uit redelijk geprijsde videorandapparatuur voor de Commodore Amiga-lijn. Genlocks, RGB- splitters, PAL-generators, computergestuurde montage- en SEG-units er komt steeds wat nieuws in Engeland, de VS en Duitsland uit. Ook de softwarehuizen laten zich niet onbetuigd. Er is ruim keuze uit diverse teken-, titel-, animatie- en SEG-pakketten. In de amateurvideo bent u voor minder dan *f* 3.000,- compleet met hard- en software uitgerust. Wie meer ambieert zal op een gemiddelde investering van *f* 5.000,- boven op de Amiga-prijs moeten rekenen.

Een schijntje in de professionele studiowereld!

Amiga ontmoet video

Zoals ik in de inleiding al aangaf is de kennismaking tussen de Amiga en homevideo nu ongeveer vier jaar oud. Alles wat de Amiga 1000 toen aan video-opties bood was meegenomen, want de VHS video- en camrecorders konden toendertijd bitter weinig. Bovendien waren de prijzen voor *camrecorders* dermate hoog dat slechts weinigen zich er eentje konden veroorloven.
Momenteel koop je voor zo'n *f* 2.400,- een prima VHSC- camrecorder die voor amateurdoeleinden een allezins redelijk beeld geeft. Low- en no-budget krijgen de kans om zelf goede videoproducties te maken. Ook de zogenaamde *montagevideorecorders* die perfecte beeldlassen maken en ook achteraf geluid en/of kommentaar kunnen toevoegen daalden beneden de *f* 2.000.
Waar komt de Amiga nu in dit plaatje voor? De Amiga 500, 1000 en 2000 zijn geschikt voor het maken van (geanimeerde) titels in meerdere kleuren, computer/business graphics, animaties, special effects en kabelkranten. Deze videofuncties kunnen ook door andere (niet-PC) videorandapparatuur vervult worden. De Amiga biedt echter de volgende voordelen:

- De vergelijkbare randaparatuur voor de Amiga is doorgaans *lager geprijsd.* Als u de Amiga al had bent u bij de studiobouw dus goedkoper uit.
- De Amiga biedt *TV-techniek.* D.w.z. de Amiga kan alles produceren wat u aan flitsende logo's, draaiende titels en geanimeerde graphics uit de kabel krijgt. De mogelijkheden van de moderne videosoftware is hier onbegrenst.
- De Amiga kan, met name in combinatie met een MC 68020 CPU, de effecten *sneller en vloeiender* dan andere animatie- apparatuur in dezelfde prijsklasse.
- De Amiga biedt meer *artistieke mogelijkheden* dan conventionele video-effectapparatuur. O.a. 3D graphics, beeldtransformaties, het digitaliseren en creatief bewerken van beelden en combinatie van computergraphics met life video.

Kortom, de Amiga biedt meer mogelijkheden in dezelfde prijsklasse. Beter kost al gauw *f* 10.000,- extra.

Computer in video-apparatuur

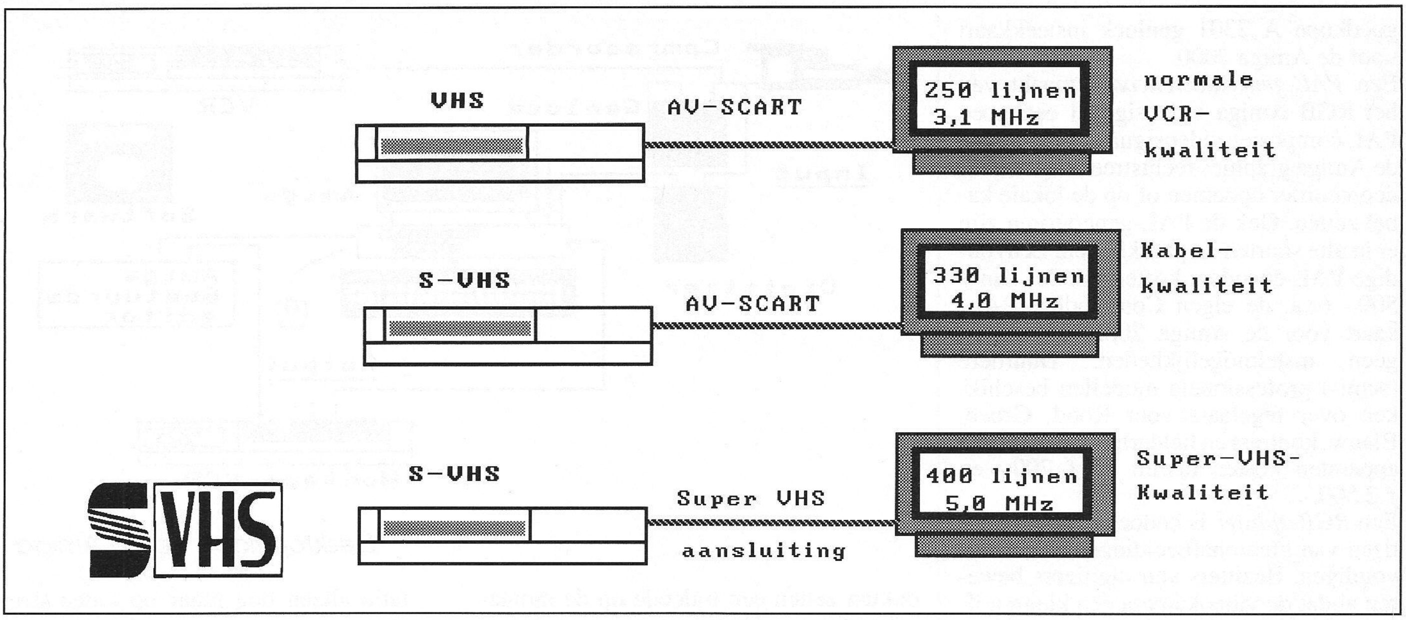
De producenten van video-apparatuur mikken de laatste jaren meer en meer op de creatieve en (semi-) professionele consument. Die wil, verwend door de TV-effecten, steeds meer. Daardoor krijgen de video- en camrecorders steeds meer computerachtige opties ingebouwd. Bijvoorbeeld titelgeneratoren met een geheugen voor meerdere pagina's in verschillende fonts en kleuren blijken steeds meer standaard op de camrecorder te worden. Overigens klopt de Amiga dit soort titelgeneratoren met glans. Niet iedereen heeft echter al de Amiga-mogelijkheden nodig. Ook de huiskamervideorecorder krijgt steeds meer snufjes. Digitalisatie van videobeelden maakt meerdere stilstaande

VHS versus Super-VHS

Er is het afgelopen jaar veel te doen geweest over *Super-VHS.* Allerlei persberichten en reklames deden voorkomen of Super-VHS het nieuwe Mekka van de videofilmer en -producent is, zodat de oude *VHS-apparatuur* in de krant kan. Dat is natuurlijk allemaal wat overdreven en de consument heeft niet zo gereageerd als de fabrikanten verwacht hadden. Eigenlijk is dat ook een beetje hun eigen schuld, want Super-VHS is momenteel nog te duur voor de kleine beurs. In kwaliteits-prijsverhouding met professionele video-apparatuur is Super-VHS echter wel een koopje!
Hoe zit dat nu allemaal met die *beeldkwaliteit?* Kritische (ex-)smalfilmers hadden het allang in de gaten. Die 235-250 horizontale beeldlijnen die de doorsnee VHS videorecorder (VCR) op de beeldbuis zet is nog slechter dan het kabelsignaal en kan niet tippen aan de 8 mm positieffilm-kwaliteit. Behalve dat lage scheidend vermogen van de VHS-beelden laat ook de kleurweergave te wensen over. Gestreepte kolbertjes met een Moiré-effect, trillende egale kleurvlakken (met name blauw en rood) en kleurstrepen bij bewegende objecten stuiten de perfectionist tegen de borst. Bij close-ups gaat het dan allemaal nog wel. Bij groothoekopnamen is het eigenlijk allemaal niet veel.
Een nog belangrijker VHS-probleem is de *kopieerkwaliteit.* Voor montage- en distributiedoeleinden dient er gekopieerd te worden. Er moeten altijd een moederband (eerste-generatie-kopie) en daarvan weer dochterbanden (tweede-generatiekopieën) gemaakt worden. Bij VHS geeft dat zoveel kwaliteitsverlies dat derde- generatie een kwelling voor het oog en de kleurenziel vormen.
Moet VHS nu naar de vuilnisbak verwezen worden? Zeker niet, want het systeem is qua apparatuur en banden heel betaalbaar, geeft tot de tweede-generatie-kopie een bevredigende kwaliteit en kan letterlijk overal afgespeeld worden. Voor de gewone amateurs biedt de moderne VHS-standaard voldoende huiskwaliteit.
Toch was de VHS-peetvader, **JVC**, gevoelig voor de argumenten van de kritische videofilmer en low- tot no-budget professional. Met in het achterhoofd dat concurrent Sony ook een kwaliteitsijzer in het vuur had staan introduceerde JVC vorig jaar de *Super-VHS- standaard.* Daarbij wordt gebruik gemaakt van twee beeldverbeterende technieken:

- Een *grotere bandbreedte.* 5 MHz tegen 3,1 MHz bij gewoon VHS biedt horizontale 400 beeldlijnen. Geen 8 mm smalfilm. Wel beter is de conventionele kabel-TV.
- Het aanbieden van een *apart helderheids (Y-)signaal* en een *apart chrominantie (= kleur) C-signaal* in plaats van een gecombineerd (FBAS) signaal maakt een aanzienlijk betere kleurweergave mogelijk.

De voordelen zijn drieledig: een hoger scheidend vermogen met meer detailscherpte (ook bij groothoekopnamen), betere kleurweergave en een zeer gering kopieerverlies. Het optimale resultaat wordt alleen bereikt als alles (kamera, [montage-]recorder en TV) van het Super-VHS-type zijn. Weergave met een Super-VHS VCR op een normale TV geeft ook een aardige verbetering van de beeldkwaliteit evenals het kopiëren van Super-VHS (moederband) naar eén gewone VHS-recorder voor de dochterkopieën.
Aan Super-VHS kleven momenteel drie bezwaren: de relatief hoge prijs voor de hardware (ook die van de Amiga!) en tapes, de aparte Y- en C-aansluiting tegen de bij VHS gebruikelijke SCART- plug, en het feit dat je Super-VHS-banden alleen in Super-VHS- recorders kan afspelen. Omgekeerd kan VHS-tape wel in Super-VHS- VCRs afgespeeld worden. Low-, no-budget professionals en serieuze video-amateurs zullen gemakkelijk over de genoemde bezwaren heen stappen. De kwaliteitswinst is de investering meer dan waard. Ook het gebrek aan Super-VHS-videorandapparatuur behoeft inmiddels geen bezwaar meer te zijn. Tal van leveranciers bieden al gekombineerde VHS en Super-VHS apparatuur en speciale S-VHS- versies.



Het oplossend vermogen in horizontale beeldlijnen bij VHS

beelden om één centraal bewegend beeld, multi-beeldweergave, paint-, pseudo solarisatie en mozaiek-effecten, stroboscoop- effect, Picture in Picture (PIP) op de TV en superpositie van titels mogelijk. Sommigen modellen kunnen tevens nog inzoomen.
Niettemin blijft de Amiga ook hier veelzijdiger en vaak goedkoper.

Waarom DTV?

Wie gaat er nu aan desktopvideo beginnen? Op deze vraag zijn verscheidene antwoorden mogelijk omdat er zo veel verschillende kategorieën gebruikers zijn. In de praktijk draait het om de volgende kategorieën videomakers:

- De *kleine professional.* Makers van low-budget bruids- en bedrijfsreportages hebben geen superprofessionele videostudio nodig. Toch wil de klant voor niet al te veel geld een leuke "aankleding" met titels, hier en daar een effectje en soms wat animaties. De Amiga biedt hier ruim voldoende mogelijkheden.
- De *Low- of no-budget AudioVisuele*-*dienst.* Openbare instellingen, het onderwijs, buurthuizen, voorlichters e.d. beschikken dikwijls niet over voldoende middelen en hebben toch behoefte aan goed in-house videowerk. De Amiga houdt hier de zaken betaalbaar en biedt in de vorm van simulaties of animaties interessante educatieve mogelijkheden.
- De *kleine kabelkrant.* Zowel voor op de lokale kabel als in-house bulletinboards. De Amiga doet het voor een fractie van de kosten.
- Last but not least de *serieuze videoamateur* die op betaalbare wijze meer van zijn/haar producties wil maken. De Amiga biedt hier een voordelige en kwalitatief goede oplossing.

Op grond van de voorgaande voorbeelden denkt u dat de Amiga niet door de "grote videojongens" en omroepen gebruikt wordt. Niets is minder waar want de logo-ontwerpers, aankondigingen makers, trucfilmers en weermannen maken volop gebruik van de Amiga. Wel is de Amiga 2000 dan gekoppeld aan topklasse randapparatuur.

De hardware

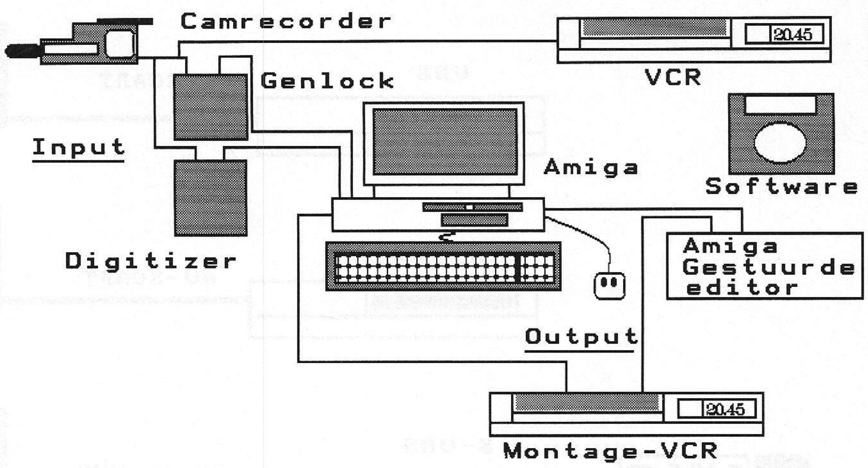
De *videorandapparatuur* voor de Amigalijn is nogal verschillend van aard en veelal in ondoorzichtelijke technische specificaties gehuld. Commodore Info geeft u hier een kort overzicht van de verschillende type videohardware en hun functie.
Een *genlock* is een apparaat dat het computervideosignaal met dat van een composiet-videobron (video- of camrecorder) kombineert tot een voor een normale videorecorder geschikt uitgangssignaal. Dat betekent dat u de op de Amiga gemaakte graphics, animaties, speciale effecten en titelpagina's aan bestaande of kameravideo- opnamen kunt toevoegen. Er is keuze om de Amiga-beelden in de voorgrond (superpositie over het onderliggende videobeeld heen) of achtergrond te gebruiken. Dat kan softwarematig of via knoppen/regelaar op de genlock.

Technisch gezien zijn er binnen het genlock twee mogelijkheden:

- De genlock zet het RGB-signaal van de Amiga om in een composiet-videosignaal en stuurt een samengesteld signaal naar de ontvangende videorecorder. De methode wordt het meest gebruikt en geeft een goede beeldkwaliteit.
- De genlock maakt eerst van het aangeboden composiet- videosignaal uit de videobron een RGB-signaal, kombineert dit met het Amiga RGB-signaal en vertaalt het gekombineerde RGB-signaal naar een composiet-videosignaal voor de ontvangende VCR. Deze methode maakt ook digitaal bewerken van het bronsignaal mogelijk. Een nadeel is een licht kwaliteitsverlies van het bronsignaal.

Behalve de hierboven beschreven heeft een genlock nog meer functies. Elke genlock is in feite ook een PAL-generator, want er komt PAL composiet-video als output uit. Duurdere (semi-)professionele genlocks zoals bijvoorbeeld de Videocomp VCG-3 bieden extra opties in de vorm van controleregelaars voor Rood, Groen, Blauw, de helderheid, kontrasten de beeldcontouren. Daarmee heeft u het eindresultaat optimaal in de hand.
Genlocks zijn verkrijgbaar als losse kastjes en insteekkaarten (alleen voor de Amiga 2000). De prijzen van genlocks variëren van circa *f* 900,- tot *f* 6.000,-. Alle waar naar zijn geld hoewel u zich bij de duurdere professionele modellen zich wel dient af te vragen of die extra kwaliteit en mogelijkheden ook daadwerkelijk lonen? Commodore zelf biedt een relatief

goedkope A 2301 genlock insteekkaart voor de Amiga 2000.
Een *PAL-generator/encoder* maakt van het RGB Amiga videosignaal een goed PAL composiet-videosignaal. U kunt dan de Amiga graphics rechtstreeks op een videorecorder opnemen of op de lokale kabel zetten. Ook de PAL-generatoren zijn er in alle soorten en prijsklassen. Eenvoudige PAL-encoders kosten minder dan *f* 500,- (o.a. de eigen Commodore PAL-kaart voor de Amiga 2000) en bieden geen instelmogelijkheden. Duurdere (semi-) professionele modellen beschikken over regelaars voor Rood, Groen, Blauw, kontrast en helderheid. Dergelijke apparaten kosten tussen de *f* 700,- en *f* 2.500,-.
Een *RGB-splitter* is bedoeld om het digitizen van kleurenafbeeldingen te vereenvoudigen. Bezitters van digitizers beweren al dat de videokamera een kleuren illustratie afzonderlijk door de filters Rood, Groen en Blauw dient te digitaliseren. Dat kost tijd en de kans op beweging is groot. Met behulp van de RGB- splitter wordt het kamerasignaal in één keer omgezet in een voor de Amiga bruikbaar Amiga-signaal. Bovendien leveren de ingebouwde electronische filters een betere beeldkwaliteit en is het bij de meeste modellen mogelijk om Rood, Groen en Blauw afzonderlijk bij te regelen. Een budget model kost ongeveer *f* 350,-.
PAL-generatoren en RGB-splitters worden steeds vaker gecombineerd tot een zogenaamde *multiprocessor*. Dit apparaat verricht beide functies en is veelal wat voordeliger dan de afzonderlijke randapparaten. De gemiddelde prijs ligt tussen de *f* 600,- en *f* 1.000,-.
De *digitizer* zet bestaande videobeelden om in voor grafische Amiga-software bewerkbare bestanden. Eenmaal in .IFF kunt u er het hele tekenpakket-arsenaal op loslaten. Een digitizer kan met een zwart-wit videokamera, camrecorder of VCR met perfect stilstaand beeld werken. Een belangrijk punt is de snelheid waarmee het digitaliseren gebeurd. Een bewegelijk object kan niet door een trage, duurt meerdere seconden, digitizer opgenomen worden tenzij u natuurlijk creatieve bewegingseffecten nastreeft. Er zijn ook apparaten die in 1/50 seconde digitaliseren en daarmee ook bewegende objecten bevriezen. De prijzen voor Amiga digitizers liggen tussen de *f* 500,- en *f* 1.000. Let bij aankoop op de aansluiting via de parallelle poort (welke Amiga!) en welke grafische beeldschermmodussen de hard- en software ondersteunen.
Enkele firma's leveren momenteel *Amiga gestuurde videomontage- apparatuur*. De Amiga functioneert hierbij als een beeld editor die de scenevolgorde en -plaats op de tape onthoudt en de montagerecorder aanstuurt. Zijn de tijd- (de meeste producten zetten een tijdcode op de montageband) en volgordegegevens eenmaal on screen ingevoerd dan verzorgt de Amiga (semi-)automatisch de montage. De prijzen voor dit soort time-code editors zijn niet mis. Reken op *f* 1.500,- tot *f* 2.000,-. Een eenvoudige losse hobby-editor is vaak goedkoper.



Desktopvideo op de Amiga

De videosoftware

In de *Amiga videosoftware* schuilt meer kracht dan in vergelijkbare titelgenerator, animator, graphics generator en SEG beneden de *f* 10.000. De Amiga biedt domweg meer voor minder geld dan de losse videorandapparatuur. Dat komt omdat de Amiga- familie al over superieure MC 68.000- en coprocessor- videotechniek beschikt en de software deze mogelijkheden er alleen nog even uit behoeft te halen.
Er is keuze uit vijf soorten videosoftware:

- *Titelsoftware* voor het maken van veel kleurige titels in meerdere fonts en kleuren, scrollende titels, titeleffecten en achtergronden. De video-titelpakketten variëren van heel basis tot TV-kwaliteit.
- *Animatiepakketten;* Animatiesoftware laat grafische figuren tussen een gegeven begin- en eindsituatie vloeiend bewegen. De Amiga neemt het benodigde rekenwerk voor zijn rekening
- *Paintpakketten* worden gebruikt voor het maken van logo's, te animeren figuren en achter- of voorgronden. Ook CAD(D)-software wordt meer en meer voor videodoeleinden ingezet.
- *Videostudio-* en *slideshow*-pakketten worden gebruikt om een reeks computerdia's of animatiesequenties aan elkaar te koppelen. De Amiga runt de komplete show. U behoeft de presentatie alleen nog maar op video over te nemen.
- *Montage- en Special Effect-software* zet de Amiga in als *beeldcutter*, *wipe*-automaat of SEQ. Kortom professioneel video's monteren met de Amiga.

Om u een indruk van de geboden mogelijkheden te geven een kort overzicht van de gangbare Amiga-videosoftware. *Titelsoftware* is er in alle maten, soorten en prijsklassen. Een basaal titelpakket doet in feite niets anders dan tekst aanmaken. De videoproducent geeft aan waar de tekst moet staan, welke kleur de letters met bijbehorende achtergrond hebben en welk lettertype er gebruikt wordt. Een aldus gemaakt titelscherm kunt u los opnemen (PAL- output) of met behulp van een genlock over eerder opgenomen videobeelden heenleggen. Hetzelfde kan tegenwoordig ook met een camrecorder die over een ingebouwde titelgenerator en paginageheugen beschikt.
Duurdere pakketten zoals TV TEXT, Videotitler en VideoPage bieden extra opties zoals meerdere fonts (ook op één titelpagina zijn meerdere fonts toepasbaar), 3D-, stroboscoop- en schaduw- effecten, het teken van standaardfiguren, beheersing van de lichtinval (op de titelletters) en shading van de achtergrond. De duurdere titelprogramma's vreten door hun mogelijkheden en dikwijls nog bijkomende animaties veel geheugen. Reken op minimaal 1 MB aan vrij RAM. De prijzen variëren van *f* 200,- tot *f* 500,-.
De *animatiepakketten* zijn bedoeld om "tekenfilmpjes" van grafische afbeeldingen en ook van titels te maken. Vroeger moest men voor een tekenfilm elk beeldje afzonderlijk (of althans de verschillende bewegingsfasen) op sheets tekenen. De Amiga wil alleen weten waar de beweging begint, waar deze eindigt en welke baan er afgelegd moet worden. Een grafi-

sche coprocessor doet de rest en bespaart u daarmee een zee van tijd.
De meeste animatiepakketten werken met een tijdbalk en een script. Door de verschillende scenes op de gewenste plaats in de tijdbalk te zetten ontstaat een lopend beeldverhaal, het script. Zit het script eenmaal in elkaar dan kan de Amiga dit als een komplete tekenfilm afspelen. Aan u de keuze of u meerdere scripts aan elkaar koppelt, de scripts afzonderlijk opneemt of hen via een genlock met life video kombineert.
Animaties zitten bij de nieuwere generaties videosoftware veelal al standaard ingebakken. Bijvoorbeeld DeLuxe Paint III, Aegis Animator (tekenpakket plus animator) en titelsoftware zoals ProVideo Plus zijn reeds van een animatiemodule voorzien.
Animatiepakketten kosten tussen de *f* 200,- en *f* 400,-.
De *Paint-software* valt uiteen in de gewone tekenpakketten zoals DeLuxe Paint III, Digi Paint of Aegis Draw en zogenaamde CADD (Computer Aided Design en Drafting) programma's. *Paintpakketten* zijn primair bedoeld als tekenhulpjes. Zij helpen de gebruiker bij het optekenen van standaardfiguren (vierkanten, lijnen, bogen, ellipsen, cirkels e.d.), arceren, het invullen van kleurvlakken, in perspectief (3D) tekenen en een aantal pakketten geeft ook controle over de lichtval. Paintsoftware wordt gebruikt voor het maken van te animeren figuren, voor/achtergronden en logo's. Behalve het geheel zelf tekenen kunnen paintpakketten tevens bibliotheekplaatjes en met een videodigitizer weggeschreven grafische files inladen. Het artistiek bewerken van gedigitaliseerde videobeelden schept een scala aan Special Effects-mogelijkheden. De meeste digitizers komen al inclusief een bijpassend tekenpakket van de leverancier. Een doorsnee paint-pakket kost tussen de *f* 200,- en *f* 350,-.
De *CADD*-software houdt zich meer bezig met de technische kant van het tekenen. In feite gaat het om een afgeleide van gewone CAD- programma's zoals die in de werktuigbouw, architectuur en electronica gebruikt worden. Bij CADD worden echter alle 3 D- effecten, wireframes (=draadmodellen), perfecte cirkels en ellipsen, standaardsymbolen, perspectieven, invul- en arceervlakken, vele nauwkeurige tekenopties en -bewerkingen ingezet voor artwork en logo's. Tevens geschikt voor het maken van sciencefiction-video's. Omdat CADD-software zoals bijvoorbeeld Aegis Draw 2000 en Dynamic CAD aanmerkelijk meer kan dan gewone paintsoftware is het prijskaartje wat hoger. Een CAD(D)-pakket kost tussen de *f* 600,- en *f* 1.400,-.
Videostudio- en *slideshow*-software hebben veel gemeen. Beide pakketten zijn bedoeld om grafische afbeeldingen en teksten te animeren tot een complete lopende voorstelling. De videostudio- pakketten zoals DeLuxe Video II, DeLuxe Productions, Video Effects 3D en Aegis Lights, Camera, Action zijn meer bedoeld voor overname op een VCR. Slideshow-software zoals Aegis Impact richt zich wat meer op stand alone Amiga diashows. Overigens zijn beide pakkettypen gewoon door elkaar te gebruiken en valt soms het onderscheid nauwelijks te maken.
Tot slot de videomontage- en SEG-software. *Videomontage*-pakketten worden doorgaans meegeleverd bij de voor de Amiga geschikte montage-apparatuur. Dit onderwerp is dermate specialistisch

Digitizers

Het bevriezen van videobeelden tot grafische Amiga-files is de laatste jaren in opmars. Met het verschijnen van gunstig geprijsde videodigitizers zoals Digiview en de komst van krachtige tekenpakketten voor de bewerking kan de Amiga-bezitter alle kanten uit. Van artwork tot creatieve videotechnieken, de Amiga met digitizer maakt het mogelijk.
Zoals bekend werkt consumentenvideo [(Super-)VHS en 8 mm video] met 25 beeldjes per seconde. Het menselijk oog ziet bij deze beeldsnelheid de afzonderlijke beeldjes als één vloeiende beweging. Een videodigitizer pakt één of meerdere beeldjes over elkaar heen, zet deze in het Amiga-RAM en de software maakt er al of niet nabewerking een .IFF-bestand van. De grafisch artiest of videomaker kunnen dit .IFF-beeld dan later met een tekenpakket bewerken. Vele digitizers laten tijdens de opname (het bevriezen) ook al softwarematige bewerkingen toe.
In principe zijn er twee digitize-mogelijkheden:

- *Het bevriezen van een stilstaand videobeeld.* Dat kan door de afspelende VCR op pauze te zetten of een (zwartwit) videokamera te gebruiken. Een probleem bij de kamera vormen de bewegende onderwerpen. Het digitizen kost al gauw enkele seconden en een beweginkje is zo gemaakt. Daar komt nog bij dat zonder RGB-splitter het nemen van niet-statische objecten onmogelijk is. In drie fasen digitaliseren duurt domweg te lang!
- *Het nemen van een "snapshot"*, de digitizer neemt een videofoto in een fractie van een seconde (bijvoorbeeld 1/50 seconde) zodat beweging uitgesloten is. De snapshot- of real-time digitizer vereist een aparte techniek en werkt prima met camrecorder en video-opnamekamera's.

We beginnen met de gewone digitizer. Dit Amiga-randapparaat is relatief goedkoop (tussen de *f* 400,- en *f* 600,-) en neemt er zoals gezegd de tijd voor. De meeste moderne digitizers leveren een goede beeldkwaliteit af. U kunt kiezen uit LOW-RES, MEDIUM- RES, HAM, (Half) Bright en Full Screen PAL (= zonder zwarte beeldrandjes). Bedenk wel dat bij de hogere reoluties 512 KB RAM te krap en 1 MB al gauw te krap wordt. Beeld kwaliteit en veel kleuren vreten vrij RAM.
Doorgaans worden de videodigitizers, al of niet via een lintkabeltje, in de parallelle poort (let op het Amiga-model!) gestoken. Op de digitizer zelf zit een tulp- of BNC-aansluiting voor de videobron. De bekendste videodigitizers in deze klasse zijn Digi View Gold, Deluxe View 4.0 en de Diamond. Voor een optimaal resultaat zijn in monochroom een zwartwit-kamera en in kleur een RGB-splitter met van een monitorschakeling voorziene camrecorder aanbevolen.
De real-time digitizer is in staat om bewegen videobeelden als een snapshot-kamera te bevriezen. Bij de Amiga in PAL worden er 50 halve beelden per seconde opgetekend en de digitizer zal in de praktijk in 1/25 seconde een volledig videobeeldje bevriezen. De prijzen voor deze snapshooters zijn aanmerkelijk hoger dan bij de conventionele modellen. Als voorbeeld twee realtime-digitizers:

- *De Snapshot Professional* (Diezemann, Kiel) met parallelle aansluiting, frame buffer met DMA, en regelaars voor kleur, helderheid en kontrast. Prijs rond de *f* 900,-. Met kleuradapter circa *f* 1.300,-.
- *De JCL Superpic* is een kombinatie tussen een komplete RGB kleurdigitizer en een genlock. Dit apparaat wordt op de RGB- poort aangesloten (de Amiga monitor door verbonden) en beschikt over alle benodigde beeldregelaars. Helaas is de prijs er ook naar. Circa *f* 1.750,- in de winkel.

Het digitizen van videobeelden wordt steeds meer een geintegreerde DTV-toepassing. De hard- en software leveren het basismateriaal voor Artwork, logo's en futuristische animaties of enerverende slideshows. Kijk bijvoorbeeld maar eens wat de verschillende omroepen er mee doen.

dat we uitleg verder aan de leveranciers overlaten. De *SEG*-software maakt wervelende grafische videoshows van .IFF- en .ANIM-files. Bijvoorbeeld spins, page-turns, strobes en confetti. Ideaal om uw videoproducties een "professional touch" te geven. SEG-pakketten zoals Aegis Animatic kosten tussen de *f* 200,- en *f* 300,-.

De video-apparatuur

De Amiga met genlock, PAL-generator, digitizer en software vormt slechts de helft van een complete videostudio. Natuurlijk zijn er ook een videokamera, een montagerecorder en wat extra randapparatuur nodig. Anders kunt u de beelden alleen op diskette vastleggen.
Er zijn twee typen videokamera's:

- De *opname-kamera* is alleen bedoeld voor het opnemen en naar een losse videorecorder doorgeven van beelden. Veel amateurs zullen een dergelijke opnamekamera niet kopen, omdat een camrecorder met ingebouwde VCR veel handiger is. Wel kan de aanschaf van een goedkope zwartwitkamera voor monochrome digitizen in aanmerking komen. Die geven een beter resultaat dan de kleuren camrecorder.
- De *camrecorder* is de video-opvolger van de oude filmkamera. Kamera en videorecorder zijn in één handig draagbaar apparaat verenigd. Er is keuze uit full-size VHS/Super- VHS-modellen, VHS/Super-VHS Compact modellen en 8 mm camrecorders. De niet Super-VHS-typen scoren wat minder op hun recorderdeel. Hun resolutie ligt tussen de 235 en 250 horizontale beeldlijnen. De kamera scoort dikwijls aanzienlijk beter, maar het ingebouwde recordertje vormt de zwakste schakel van de videoketen. Hetzelfde geldt ook voor de 8 mm-typen, waarvan wel het geluid beter is dan gewoon VHS C. Super-VHS camrecorders, met name de modellen met een brede (62 mm) koppentrommel, scoren qua oplossendvermogen beter dan de conventionele TV.

Voor het monteren van videotape zijn altijd twee videorecorders (VCRs) nodig. Knippen kan niet. Wel overschrijven van beelden van de ene naar de andere recorder. Heeft u al een camrecorder dan is de aanschaf van een *montage-VCR* of het gebruik van de huiskamer-video voldoende. Die montage-VCR verdient door zijn exacte beeldlassen en extra geluidsmogelijkheden de voorkeur. Dit type apparaten "vreet" de te monteren scenes zo min mogelijk aan (gewone videorecorders wissen gewoon een seconde of wat uit en kwakken daar de volgende scene overheen!) en zorgt voor storingsvrije lassen. Met een flying erase-kop zijn ook naadloze inserts op een al gemonteerde tape mogelijk. Via de Audio-Dub kunt u achteraf nog geluid toevoegen en via de Audio-Mix het bestaande stereogeluid opnieuw met kommentaar of achtergrondmuziek mengen. Verder kunt u de Amiga natuurlijk gebruiken voor het creëeren van geluidseffecten en electronische muziekjes.

VideoComp

Het duitse bedrijf VideoComp is een specialist in DTV-apparatuur voor de Amiga. Er is een ruime keuze uit amateur, semiprofessionele en professionele randapparatuur. Daarnaast heeft deze firma een uitgebreid DTV-softwarerepertoire. Wij bespreken hier enkele voor de hobbyist en low-budget videoproducent interessante apparaten.
De VCG-3P is een genlock van ware studiokwaliteit. De bandbreedte is omschakelbaar tussen 7,5 en 5,5 MHZ (voor Super-VHS). Er zijn fading-, superimpose- en invert-opties. Voor het stand alone-gebruik beschikt de VCG-3P over een blackburstgenerator. Een interne klokoscillator zorgt voor exacte synchronisatie en voorkomt dat het beeld bij een slecht signaal ineenstort. En verder zijn er regelaars voor RGB, contourkontrole en helderheid. Het apparaat is geschikt voor VHS, 8mm en Super- VHS.
Zoals gebruikelijk bij de meer professionele videorandapparatuur is de VCG-3P genlock van een afstandbediening met verbindingskabeltje voorzien. Dat voorkomt allerlei onhandig gewriemel achter of naast de Amiga. U kunt nu overal comfortabel bij, vooral als er meerdere videoapparaten tegelijkertijd bediend moeten worden. Helaas kost al dit moois wel eventjes rond de *f* 4.500,-.
Goedkoper is de semiprofessionele VCG-3 genlock-interface voor de Amiga 500 en 2000. Voor rond de *f* 2.200,- krijgt u een genlock die in een doorsnee videostudio niet zou misstaan. Er is nauwelijks signaalverlies (een probleem bij vele goedkopere modellen) en u kunt de kleur (RGB), het kontrast, de contouren (links en rechts), de luminantie en de fade-time regelen. Wat slordig vonden wij de externe voeding. Inmiddels is er ook nog een VCG-3/R versie leverbaar met mogelijkheid tot afstandsbesturing en triggering via een video-editorunit. Binnenkort wordt ook de Super-VHS-versie van deze VCG-3 genlock verwacht.
De VideoComp VCW-1 is een kwaliteits PAL-generator voor Amiga RGB. Op het apparaat zijn de kleuren Rood, Groen, Blauw, de helderheid en het kontrast afzonderlijk regelbaar. Er zijn twee versies waarvan het rond de *f* 700,- kostende /AC-model over een interne netvoeding beschikt. Het gewone rond de *f* 640,- kostende model snoept de stroom bij de Amiga weg. Voor informatie: VideoComp, Begerstrasse 193, 6000 Frankfurt 60.



De VideoComp VCW-1

PBC

Achter de naam PBC gaat de duitse firma Peter Biet Computerdesign schuil. Ook dit bedrijf richt zich speciaal op Amiga video- apparatuur en levert tevens versnellerkaarten, digitizers en harddisks.
Wie een professionele genlock voor beneden de *f* 4.000,- zoekt kan terecht bij de door Peter Biet verbouwde Commodore A2301 genlock, de PAL-RGB-Videotoaster. Biet voegde er een afstandsbediening, een geintegreerde real-time digitizer, een geintegreerde Super-VHS editor, en verbeterde electronische schakelingen aan toe. Nu kunt u ook de helderheid, het kontrast, de kleurverzadiging, impose, mix en triggering van de in de Amiga geplaatste kaart op afstand regelen. De volledige set kost circa 3.500 mark. Als optie is er ook nog een additieve keyframe stanzer. Wie alleen de genlock wil is voor ongever *f* 1.100,- klaar. Jammer is dat een Super-VHS-aansluiting op de genlock zelf (vooralsnog?) ontbreekt. Wel is er een Super-VHS naar RGB- converter leverbaar.

Een lichte teleurstelling vormt altijd de overname van het Amiga RGB-beeld op standaard-VHS. De kwaliteit van het opgenomen beeld is aanmerkelijk slechter dan het Amiga-origineel. Het kwaliteitsverschil bij het kopiëren werd reeds uitvoerig toegelicht. Bij Super- VHS krijgt u een veel beter resultaat. Ook hier geldt dus weer alle waar naar zijn geld. Super-VHS is circa twee maal zo duur. De prestaties betalen deze extra investering echter vanzelf terug.
De Amiga geeft RGB-af en Super-VHS werkt met gescheiden C- en Y- signalen. Met behulp van een RGB-Converter zijn Super-VHS- signalen naar het door genlocks en andere Amiga video-apparatuur gebruikte RGB-signalen vrijwel zonder kwaliteits verlies te vertalen.
Voor consumenten-video zijn er tegenwoordig ook allerlei (semi-)professionele videocorrectors en SEGs te koop. Een *videocorrector* maakt nabewerking van het videosignaal mogelijk. Bijvoorbeeld het bijwerken van een kleurzweem, lichter/donkerder, het bijregelen van het kontrast, het wat opkrikken van de scherpte, fade in/out en geluidsmixages. De SEGs verzorgen wipes, PIPs, paint, mozaiek, zwart-wit, reverse en geluidsmixages.

Kabeltjes

In de videostudio ontkomt u niet aan een warboel van kabeltjes. Er zijn zoveel aansluitingen in gebruik dat de leek al snel door de bomen het bos niet meer ziet. De meest gangbare standaard-VHS videoplug is de *SCART*-stekker. Deze gekombineerde beeld & geluidstekker past zowel op een gewone videorecorder als de Amiga- videouitgang. Alleen gebruiken diverse videorandapparaten voor de Amiga zoals genlocks en digitizers weer een BNC-plug. Camrecorders beschikken doorgaans over een kwetsbare mini videojack (10 of 12-pins) en zelden over een SCART-aansluiting. Dat betekent al twee verloopjes. Dan zijn er ook nog de tulppluggen voor audio-of videoaansluiting en niet te vergeten de C- en Y-aansluiting voor Super-VHS. Kunt u het nog volgen?
Gelukkig kan een goedgesorteerde videospecialist u aan de meeste verloopjes helpen. Daar zit helaas een stevig prijskaartje, reken op *f* 40,- tot *f* 50,-, aan vast. Bij meerdere verloopjes gaat dat flink in de prijzen lopen. In de praktijk blijken amateurkabeltjes behoorlijk kwetsbaar. Koop er daarom een paar extra van of schaf een 10 Watt soldeerbout aan. Zelf kabeltjes aanmaken kan zeer goed lopen mits u terdege weet hoe zij gesolderd dienen te worden. Kontroleer altijd de bedradingsschema's anders brandt de boel door!
De Amiga en de (semi-)professionele VHS-studio groeien steeds meer naar elkaar toe. Computer en video raken geintegreerd tot één veelzijdig medium waar slechts de fantasie de grenzen stelt. Toegegeven, niet iedereen zal een videorecorder, genlock, digitizer en kamera aan zijn/haar Amiga hangen. De low- tot nobudget videoproducent en de ware videohobbyist zullen de Amiga mogelijkheden echter best weten te ontginnen. En dat voor een budget waarvan u bij lange na geen videostudio kunt kopen.

U.S.

De *Digisplitt* is, zoals de naam al doet vermoeden, een duitse RGB- splitter. Deze RGB-splitter beschikt over signaalverbeterende circuits, regelaars voor RGB, kleur, kontrast en luminantie plus kleurechte monitorkontrole van het bereikte resultaat. De Digisplitt kost los *f* 700,- en met de DigiView Gold digitizer rond de *f* 1.000,-.
Voor informatie: Peter Biet Computerdesigns, Dieterhausenerstrasse 28, 6409 Friesenhausen.



LAMM Genlock interface

Lamm Computersystemen

De duitse in Amiga DTV gespecialiseerde firma Michael Lamm Computersystemen is in Nederland nauwelijks bekend. Dat is jammer, want dit bedrijf biedt een zeer ruime keuze uit hobby- en (semi-)professionele DTV-apparatuur. Wij zullen hier kort enkele randapparaten bespreken.
De Super-VHS-versie van het Studio Genlock-Interface werd speciaal voor de kleine Super-VHS-studio ontworpen. Deze genlock beschikt over de volgende specificaties:

° Bandbreedte 6,2 MHz tot 8,0 MHz en standaard Super-VHS
° Resolutie maximaal 625 horizontale beeldlijnen
° PAL-systeem, 4,43 MHz met kleurdraagsignaal 8 MHz
° Signaal-ruis-afstand, best 86 dB bij S-VHS
° Aansluitingen BNC (standaard VHS en S-VHS), preview en DB-23 RGB
° Eigen voeding
° Geschikt voor alle Amiga-modellen
° Eigen oscillator-circuit voor beeldkorrektie
° Effecten; superimpose,, inverse, verrekijker-effect, fades
° Ondersteunt elke grafische Amiga-modus
° VTR beeldstabilisatieschakeling

De prijs van deze VCG S-VHS genlock zal rond de *f* 3.500,- komen te liggen.
De oudere VHS-versie van de Studio VCG genlock kost rond de *f* 2.000,- en geeft een wat mindere kwaliteit. Beter is de broadcast-uitvoering van rond de *f* 3.000,- die geschikt is voor TV-kwaliteit. Dit apparaat heeft o.a. een 12-traps fasenregelaar en een Sony Sync. compatibele DSK-uitgang.
De ML LAMM RGB-splitter is geschikt voor S-VHS en VHS. Er zijn regelaars voor Rood, Groen en Blauw, plus schakelaars voor S- VHS/VHS en Video/Amiga (voorkomt het lastige gewissel van videokabeltjes).
Specificaties:

° S-VHS t/m 8 MHz
° Aansluitbaar op RGB-monitor
° Resolutie maximaal 625 beeldlijnen horizontaal
° Geschikt voor kabelgebruik
° BNC-aansluitingen voor video

Prijs circa *f* 380,-.
De Prolock-HV is een amateur-genlock-interface voor de Amiga 500, 1000 en 2000. Deze genlock beschikt over een monitor RGB-aansluiting aan de voorzijde, video in/out BNC aan de voorzijde en regelaars voor superimpose, fade in/out en reverse.

Overige specificaties:

° Bandbreedte 6 MHz

° PAL-systeem 4.43 MHz met gekoppelde 8 MHz kleurdragersignaal

° 4096 kleuren

° VTR beeldstabilisatie

° Blue-Box-Keying voor de Amiga-achtergrond

Deze Prolock HV series II genlock kost met Aegis Images rond de *f* 1.250,-.
Voor informatie: Michael Lamm Computersysteme, Schönbornring 14, 6078 Neu-Isenburg 2.

Merkens E.D.V.

E.D.V. staat voor Elektronische DatenVerarbeitung waaronder ook Amiga DTV valt. En daarvoor heeft Barbara Merkens voldoende videorandapparatuur in voorraad. Als eerste de digitizers. De VD3 is een 20 ms real-time digitizer voor de Amiga 500, 1000 en 2000. Dit handmatige model digitaliseerd in zwartwit en kan een grijstrap van 16 tinten aan. Via de outline-modus kan de gebruiker beeldpartijen van een gelijke helderheid omranden. Lopende beelden kunnen met deze digitizer prima in zwartwit bevroren worden. In de duurdere VD4-versie is een RGB-splitter ingebouwd. Stilstaande beelden kunt u zondermeer in RGB digitaliseren. Voor het in kleur digitaliseren van bewegende beelden is een videorecorder met een RAM-geheugen voor digitale beeld-opslag nodig. Deze laatste truc is niet meer nodig bij de VD 2000 Amiga real-time color Digitizer. Behalve de kombinatie digitizer en RGB splitter is de VD 2000 tevens geschikt als "grafische kaart" voor de Amiga. Deze digitizer is geschikt voor alle grafische Amiga- modussen, kan de data opslaan in .IFF en ruw dataformat, en kan zowel werken met Bit-Planes als pixels. Regelbaar zijn het kontrast, de helderheid, horizontale en vertikale oscillator plus de outline en differentbeeld-modus. Jammer is dat u voor al dit fraais wel eventjes 3.000 piek dient neer te tellen.
Merkens EDV verkoopt vijf verschillende genlocks. De goedkoopste is de mini Gen voor net boven de *f* 400,-. Deze mini genlock is een langwerpig kastje dat u in de RGB-poort van de Amiga steekt. Bezwaarlijk zijn de uitsteek naar achteren en het feit dat de RGB-monitor niet doorverbonden kan worden. Aan de andere kant komt alleen composietvideo uit de genlock. De beeldprestaties zijn redelijk en geschikt voor aftiteling en RGB naar FBAS-conversie. Het apparaatje past op alle leden van de Amiga-familie.
De door Merkens geleverde AG 5 lijkt sprekend op de Rendale A8802 genlock. Er zijn geen uitwendige bedieningsmogelijkheden en de beeldkwaliteit is redelijk tot goed. De bandbreedte ligt tussen 4 en 5 MHz. Ook deze genlock gaat in de Amiga RGB-poort. Nu kan de monitor echter wel doorverbonden worden. De AG 5 past op de Amiga 500, 1000 en 2000 en kost minder dan *f* 1.000,-.
De AG6 genlock is een Amiga 2000-model die perfect onder de systeemkast past. Deze genlock valt in C, BASIC of machinetaal te programmeren en beschikt over 11 in/uitgangen. De software kan ook een beeldplaatafspeler aansturen. Inmiddels is de AG-7 opvolger aangekondigd, maar daar hebben we nog geen specificaties van.
De Maxi-Gen is met een prijskaartje van rond de *f* 3.000,- de duurste Merkens Genlock. Dit professionele externe genlock wordt onder of de systeemkast van de Amiga 2000 geplaatst. De Maxi-Gen beschikt over de volgende specificaties:

° Bandbreedte 5,5 MHz

° 13 in/uitgangen waaronder video-in/out, RGB-monitor out, RGB BNC-out, Sync-out, RGB Chromakey-out, Keyfill-out, Black-Burst-in, en Blanking-in.

° RGB- en Video-mix

° Broadcasting beeldkwaliteit

° Regelaar voor o.a, Rood, Groen, Blauw, luminantie, contouren en kontrast.

Voor informatie en prijzen; Merkens EDV, Fuchstanzstrasse 6A, 6231 Schwalbach.

Genlock Magni 4005

Wie voor rond de *f* 4.800,- een professionele Amiga 2000-genlock (allen voor type B) zoekt kan bij Altycos (tel.: 079-510757) terecht. Daar importeert men de Magni 4005 die reeds door een aantal TV-stations en omroepen met vrucht gebruikt wordt. Wat krijgt u nu voor zo'n 5.000 gulden aan genlock in huis? In de eerste plaats een keurige insteekkaart voor een vrij Amiga 2000- slot. Natuurlijk is deze genlock geheel studiosynchroon en levert de kaart een prima composietvideo-output. Over de precisie en kwaliteit zal niemand te klagen hebben. In de praktijk kan de Magni met alle gangbare videosignalen, ook die van een matige kwaliteit, uit de genlock-voeten. Dat komt door de zeer hoogwaardige sync-generator die bijspringt waar het externe referentiesignaal het laat afweten! Verder zijn alle gangbare professionele schakelingen en regelaars aanwezig. Aan de achterzijde van de genlock-kaart zitten de dipswitches voor de keuze van de arbeidsmodus.
De montage is eenvoudig. Het genlock-deel gaat in een vrij slot van de Amiga 2000 type B. De bijbehorende videokaart in het Amiga videoslot. Desgewenst kunt u nog een afstandsbediening met o.a. schuifregelaars voor de luminantie-keying en traploze superimpose, schakelaars voor invert-keying automatisch overvloeien en fading aanschaffen. Kortom, de Magni 4005 is zijn prijs ten volle waard. Voor de amateur zal de prijs waarschijnlijk te hoog uitvallen.
Overigens heeft Altycos een groot aanbod van genlocks, videotoasters en digitizers in de prijsklasse van *f* 500,- tot *f* 6000,-. Vraag om de katalogus, tel.: 079-510757.

Commodore zelf

Van Commodore Nederland hoor je eigenlijk niet zo veel over DTV voor de hobbyist op de Amiga. Dat laat men liever aan de eigen professionele Amiga-afdeling of in DTV gespecialiseerde bedrijven over. Toch kunt u bij de Commodore dealer de hoogwaardige A 2301 en Amiga PAL-kaart bestellen. De prijs valt zondermeer redelijk te noemen. Jammer is dat op de rond de *f* 400,- kostende genlock geen regelaars voor fading, RGB e.d. zitten. Daarvoor moet u, zoals in het voorgaande beschreven, bij de duitse heer Biet zijn.
Volgens het geruchten-circuit staat er een komplete desktopvideo- module, het DTV-systeem, voor de Amiga 500 op stapel. In een externe behuizing zitten de A 2301, een RGB-splitter en Diamond digitizer. Over de winkelprijs en introductiedatum was op het moment van schrijven nog niets bekend. Voor informatie: 020- 882222.

Catronix

Last but not least de nijvere nederlandse importeur en leverancier van Amiga-randapparatuur Catronix. Hoewel de nadruk niet zoals bij de duitse leveranciers, op DTV ligt verkoopt deze firma in deze toch leuke dingen. Bijvoorbeeld de Rendale A8802 genlock, een aantal digitizers en DTV-software. Voor informatie: 010-4507696.

Tabel DTV-Software

3D Graphics	Paint 3D-objecten	129	Altycos
Aegis 3D Modeller	3D animaties + tekenen en raytracing	279	Altycos
Aegis Animator met Images	Animatie- met paintprogramma	399	Altycos
Aegis Draw	Paint-pakket	329	Altycos
Aegis Draw 2000	CAD(D)	679	Altycos
Aegis Impact	Business Graphics presentatie	249	Altycos
Animate 3D	3D-animaties raytracing	549	Altycos
Animation Effects	Wipe, scroll en flipanimaties	129	Altycos
Animation Stand	Animatieprogramma	129,50	Altycos
Broadcast Titler	Pro-titelgenerator	?	Videocomp
Caligari	Pro-animatiepakket met raytracing	?	Videocomp
Deluxe Paint II	Paint-pakket	249	div. lev.
Deluxe Paint III	Paint- en animatiepakket	299	div. lev.
Deluxe Productions	Videopresentaties	499	Altycos
Deluxe Video	Videopresentaties	249	div. Lev.
Digipaint	Paint-pakket	149	Altycos
Dynamic CAD	CAD(D)	1399	Altycos
Forms in Flight	Animatie 3D draadmodellen	199	div. lev.
Graphic Studio	Tekenpakket, artwork	159	Altycos
Lights!, Camera!, Action!	Kompleet animatie- en presentatieprogramma	229	Altycos
Logic Works 2.0	CAD(D)	729	Altycos
Master Video Page	Titelgenerator	440	div. lev.
Page Flipper	Tekstanimaties	129,50	Altycos
Photon Paint	Paint-pakket	249	Altycos
Photon Videocel animation	Animatiepakket	369	Altycos
Pro Draw	Artwork	529	Altycos
ProVideo CGI	Videotitels Profes.	825	Altycos
ProVideo Plus	Professionele titels	625	Altycos
Sculpt 3D	3D tekenprogramma	249	Altycos
Sculpt 4D	3D tekenprogramma en 3D animatie	1199	Altycos
TV Show	Presentatieprogramma	249	Catronix
TV Text	Video-aftiteling met pro-achtergronden	249	Catronix
The Caligrapher	Calligrafische titels	325	Altycos
The Director	Videopresentatie	179,50	Altycos
Turbo-Silver	3D teken- en animatie-	729	Altycos
Video Effects 3D	3D-animaties van tekst, logo's en grafieken	489	Altycos
Video Generic Master	Titelgenerator	300	Kimatek
Video Wipe Master	Overvloei-effecten voor 2 videobronnen. SEG.	300	Kimatek
Videoscape	Raytracing animatie	399	div. lev.
Videotitler	Titel-animatieprogramma	299	div. lev.
Zuma Fonts	Titel-fonts	97,50	div. lev.

Amiga Nieuws in het kort

Amiga met 68030

Alhoewel Commodore op de Comdex tentoonstelling in Las Vegas nog geen Amiga 3000 wenste te annonceren (een machine die al in 1988 in Hannover voor het eerste werd genoemd) werd er wel een Amiga 2500 met de 68030 coprocessor op 25 MHz gelanceerd. De 2500/30 heeft een mathematische comprocessor (25 MHz 68882) en een memory management unit (68851) aan boord evenals 2 MB 32-bits geheugen en 1 MB 16-bits RAM, een 3,5 inch diskdrive en 40 MB harddisk. De Amiga 2500/30 is volgens de Amerikanen te duur en vooral bedoeld voor de verticale markten in Europa (zoals de video-toepassingen e.d.).
Liefhebbers met een Amiga 2000 kunnen naar analogie met de 68020 van de gewone A2500 met de A2630 uitbreidingskaart hun 2000 upgraden naar de 68030.
De ontwikkelingen voor de Amiga gaan volgens Amiga product manager Keith Masavage echter niet geheel aan de 'gewone' Amiga gebruiker voorbij omdat er hard wordt gewerkt aan een belangrijke update van het besturingssysteem. Deze versie wordt in de VS begin volgend jaar verwacht. Rond die tijd zal ook de Amiga 3000 waarschijnlijk worden gelanceerd, omdat Commodore volgens Masavage in ieder geval wacht tot Unix System V release 4 beschikbaar is, en dat zal wel in de eerste helft van 1990 zijn. Het zal dus wel de traditionele grote Commodore persconferentie in Hannover worden.
De Comdex zag ook de introductie van nieuwe versies van de auto-booting harddisk controllers/geheugenuitbreidingskaarten A2091 en 2091-40

De flickerFixer

Iedereen die het Amiga 2000 monitorbeeld eens nader bekeken of gefotografeerd heeft zal zijn opgevallen dat het beeld flikkert of scanlijnen vertoont. Dat ligt niet aan de Amiga zelf, maar aan de videokaart. Nu blijft het altijd de vraag in hoeverre zoiets echt storend is. Het menselijk oog heeft bij de doorsnee Amigatoepassingen weinig last van dat nauwelijks waarneembare geflikker. Bij grafische, desktopvideo, desktoppublishing en fotografische (van het beeldscherm) toepassingen kan de interlace-flikker wel storend worden.
De AGA 2000-kaart van Micro Way verhelpt de interlace-flikker en wordt daarom de "flickerFixer" genoemd. Een heel toepasselijke naam, want met deze insteek-videokaart krijgt u een flikkervrij 704 x 470 pixels overscanbeeld in de 4096 paletkleuren van de Amiga. Deze lust voor het grafisch oog is geschikt voor de populaire VGA- en Multiscan-monitoren (O.a. de NEC Multisync en de Mitsubishi XC1429C). Natuurlijk werkt de AGA 200 flickerFixer met alle bestaande grafische Amiga-software.
Het enige bezwaar is de stevige prijs die rond de *f* 1.400,- ligt. Info: Micro Way, 32 High ST., Kingston-Upon Thames, Engeland.

Geruchtenmachine

In een aantal buitenlandse bladen zijn de laatste maanden speculaties opgedoken over een mogelijke nieuwe Commodore 64. Commodore baas Irving Gold zou hierover in een toespraak iets hebben losgelaten.
De Commodore 64-III of de Commodore 64-GS, zoals hij al in suggestieve verhalen werd genoemd, zou gebruik maken van de 65816 processor (16 bits op 4 MHz) die in staat is om de commandoset van de 6510 van de oude C64 te verwerken. Daarnaast zou de nieuwe C64 een 128 KB geheugen, betere grafische en geluidsmogelijkheden en een ingebouwde 3,5 inch diskdrive hebben. De machine zou in de prijsklasse van 600-900 gulden moeten komen te liggen. Het lijkt er bijna op alsof iemand de geruchten voor de introductie van de C128 nog eens dunnetjes overdoet. Gezien de grote voorkeur voor Commodore Nederland voor de Amiga en voor het zakelijke computersegment lijkt een introductie van een C-64/128-II-IGS nauwelijks te verwachten.

WordWriter 4

Er is een nieuwe versie van de bekende tekstverwerker WordWriter voor de C64 van softwaremaker Timeworks. Het gebruikersinterface is verbeterd en er zijn meer fonts, mini-graphics, tekst-layout en betere afdrukresultaten ana het programma toegevoegd.
Versie 4 heeft nu 8 ingebouwde lettertypes in groottes van 9 tot 72 punten, alle met 5 speciale effecten. Naast lettertypes zitten er in het pakket 90 mini-tekeningen.
Naast het standaard pakket is er een uitbreidingspakket beschikbaar met 16 extra lettertypes.

Media Line

Niet iedereen is een grafisch artiest en het maken van achtergronden op de Amiga kost naast inzicht ook veel tijd. Free Spirit software doet in deze een aantal aardige voorzetjes in de vorm van tekst-achtergronden en animatiescenery. In het pakket Media Line Video & Animation zitten twee diskettes met 30 afbeeldingen voor computer-slideshows, businesspresentaties, video- en kabel-TV. De beelden zijn in het PAL 352 x 552 overscan-formaat zodat u hen direkt van de Amiga op tape kunt overnemen. Het gebruikte kleurenpalet is de VideoScape 32 kleurenstandaard. Ook is weergave in de HAM-modus mogelijk.
De eerste diskette bevat achtergronden voor teksten en wat scenery-voorbeelden. Op de tweede diskette staan structuren en High-Tech-platen. Allen goed bruikbare afbeeldingen. Het blijft natuurlijk altijd de vraag of u er iets mee kunt. Wat de een mooi vindt is voor de ander niet geschikt. Daarom raad ik u aan deze diskettes eerst te bekijken alvorens hen voor circa *f* 80,- aan te schaffen.

Media Line Clipart

Enkel tekst blijft ook bij desktoppublishing een saai gebeuren. Een plaatje er bij maakt de tekst leesbaarder en streelt oog plus geest. DTP-plaatjes kan men zelf tekenen met een paintpakket, met een scanner of digitizer van een bestaande afbeelding naar het Amiga-RAM kopiëren of uit een plaatjesbibliotheek laden. De Media Line Clipart-disk van Free Spirit software is zo'n plaatjesbibliotheek.
Net als bij andere gelijksoortige beeldbibliotheken bevat deze Clipart-schijf zwartwit plaatjes over veel gebruikte onderwerpen. Bijvoorbeeld over:
Feestdagen; kerstmis, kadootjes, engel, drie koningen, nieuwjaar, klokken en een kaars
Mensen; vrienden, moeder, filmkijkers, student, feestgangers, gedigitaliseerde filmsterren en bijbelverkopers
Sprookjesfiguren; heks, tovenaar
Sporten; diverse sporters
Vlieg- en voertuigen; van zeilboot en stoomtrein tot spaceshuttle en helicopter
Voedingsmiddelen; kreeft, pizza, fast food, dranken, groenten
Dieren; olifant, leeuw, tijger, poes, e.d.
Infanterie met tanks
Dit clipartpakket ligt bij uw softwaredealer voor rond de *f* 80,-.


Charlie Chip's
SOFTWIRWAR
DOOR:
CHARLIE.. DOE ME 'N LOL ZEG.. BEN JE HIER NOU NIET WAT TE OUD VOOR?!
ZIE GINDS KOMT DE STOOMBOOT
..IK BEDOEL.... BRIEFJES AAN SINTERKLAAS!
ZINGEN...
SINTERKLAASJE KOM MAAR BINNEN MET JE KNECHT..
VIND JE JEZELF NOU NIET ERG KINDERACHTIG!
NOU.. WAT IK VRAAG IS ANDERS BEPAALD NIET KINDERACHTIG!..
EEN MAC II VAN APPLE!!
DÁÁR WIL IK WEL EVEN VOOR ZINGEN..
WEL ALLEMACHTIG!
EN... HEB JE'M GEKREGEN?!! JE MAC II ?!
JA..
..MAAR NIET VAN APPLE!..
HMM.. MARSEPIJN!! TJA.. WAT VERWACHT JE ANDERS VAN SINTERKLAAS!..